

**UNIVERSIDADE FEDERAL DE SERGIPE  
CAMPUS ALBERTO CARVALHO  
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

**SUELLINGTON MIGUEL SANTOS**

**PROJETO E IMPLEMENTAÇÃO DE UMA  
ARQUITETURA ROLAP PARA AUXILIAR O ENSINO DE  
FERRAMENTAS DE SUPORTE À DECISÃO**

**ITABAIANA  
2018**

**UNIVERSIDADE FEDERAL DE SERGIPE  
CAMPUS ALBERTO CARVALHO  
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

**SUELLINGTON MIGUEL SANTOS**

**PROJETO E IMPLEMENTAÇÃO DE UMA  
ARQUITETURA ROLAP PARA AUXILIAR O ENSINO DE  
FERRAMENTAS DE SUPORTE À DECISÃO**

Trabalho de Conclusão de Curso submetido  
ao Departamento de Sistemas de  
Informação da Universidade Federal de  
Sergipe, como requisito parcial para a  
obtenção do título de Bacharel em Sistemas  
de Informação.

**Orientador: Msc. ANDRÉ VINICIUS RODRIGUES PASSOS NASCIMENTO  
Coorientador: Dr. METHANIAS COLAÇO RODRIGUES JÚNIOR**

**ITABAIANA  
2018**

SANTOS, Suellington Miguel.

Projeto e implementação de uma arquitetura ROLAP para auxiliar o ensino de ferramentas de suporte à decisão / Suellington Miguel Santos – Itabaiana: UFS, 2018. 79f.

Trabalho de Conclusão de Curso em Bacharel em Sistemas de Informação – Universidade Federal de Sergipe, Curso de Sistemas de Informação, 2018.

1. Ferramenta de suporte à decisão.
2. Sistemas de apoio à decisão.
3. Sistemas de Informação.

I. Projeto e implementação de uma arquitetura ROLAP para auxiliar o ensino de ferramentas de suporte à decisão.

**SUELLINGTON MIGUEL SANTOS**

**PROJETO E IMPLEMENTAÇÃO DE UMA ARQUITETURA  
ROLAP PARA AUXILIAR O ENSINO DE FERRAMENTAS DE  
SUPORTE À DECISÃO**

Trabalho de Conclusão de Curso submetido ao corpo docente do Departamento de Sistemas de Informação da Universidade Federal de Sergipe (DSIITA/UFS) como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Itabaiana, 22 de maio de 2018.

**BANCA EXAMINADORA:**

---

Prof. André Vinicius Rodrigues Passos Nascimento, Mestre  
Orientador  
DSIITA/UFS

---

Prof. Methanias Colaço Rodrigues Júnior, Doutor  
Coorientador  
DSIITA/UFS

---

Prof. José Aélío de Oliveira Júnior, Doutor  
DSIITA/UFS

## **DEDICATÓRIA**

Dedico este trabalho aos meus pais, João José e Irailde Miguel, que sempre estiveram dispostos a me darem a oportunidade de conseguir concluir o curso e por sempre estarem em primeiro lugar na torcida por mim, em tudo. Tenho certeza que sem esse pilar eu não teria conseguido alcançar esta etapa. E isso, é só um pouco da quantidade de coisas que eu poderia agradecer, não existem palavras para o quanto eu gostaria de falar, obrigado por tudo que fizeram e que ainda fazer por mim. Amo vocês.

Também agradeço à minha irmã Suelen, que sempre esteve ao meu lado me ajudando de sua maneira sempre que precisei. À minha namorada Alice Barreto, que também sempre esteve ao meu lado, me dando forças para continuar. Aos meus amigos e amigas, a todos os familiares, em especial para meus tios Jailton e Cilene, minha prima Ingrid e a minha avó, Dona Jove, que já se foi, mas que sempre torcia por mim. Enfim, a todos que torcem por mim, meu muito obrigado.

## AGRADECIMENTOS

Agradeço aos meus pais por sempre darem o seu melhor em relação a mim, permitindo que eu sempre tivesse uma boa educação, fazendo com que eu conseguisse alcançar essa etapa. Não tenho palavras para demonstrar o quanto sou grato por tudo que vocês fizeram e fazem por mim, meu muito obrigado. Amo vocês!

Agradeço também à minha irmã Suelen, saiba que você tem uma participação muito importante nessa minha conquista. Também agradeço à minha namorada Alice, que sempre esteve comigo quando precisei, me dando forças e incentivos para que eu conseguisse chegar ao fim. Amo você!

Agradeço a todos os meus familiares que sempre estiveram na torcida por mim. Em especial, agradeço aos meus tios Jailton e Cilene, juntamente à minha prima Ingrid e à minha avó Dona Magnólia, que sempre me deram total apoio para que eu conseguisse alcançar esse objetivo. Muito obrigado por fazerem parte da minha vida, me ajudando e me apoiando em tudo que foi possível.

Quero agradecer também a todos os meus amigos. A todos os grandes amigos que conquistei no Monteiro, em especial a Patrícia e Sidieres. Aos meus amigos de infância, em especial, Jadson Santos, Matheus, Jéssica e Jefferson. Aos amigos da UFS, que me acompanharam em algum momento no decorrer dessa longa caminhada, em especial a Renan, Alcymar, John, Iana e Luiz. E também aos meus grandes amigos, Jadson Ribeiro e Renato Caetano, vocês foram essenciais em tudo que aconteceu nessa jornada, me dando apoio e xinga sempre que precisei, vocês foram fundamentais. E também gostaria de agradecer a todos os conhecidos que influenciaram de alguma forma para o alcance dessa conquista.

E por fim, quero agradecer a todos que fazem parte do departamento de sistemas de informação, sem exceções, técnicos, secretários e a todos os professores, obrigado por terem contribuído com a minha formação com seus conhecimentos e seus ensinamentos durante as aulas. Em especial, ao meu orientador André Vinicius por ter sido paciente e dedicado seu tempo me ajudando com todo o seu conhecimento, fazendo com que tudo isso se tornasse possível, cada conversa, cada e-mail trocado e cada reunião foram fundamentais não só para que cada ponto desse trabalho fosse concluído, mas também foram essenciais para a minha formação. Muito obrigado!

SANTOS, Suellington Miguel. **Projeto e implementação de uma arquitetura ROLAP para auxiliar o ensino de ferramentas de suporte à decisão**. 2018. Trabalho de Conclusão de Curso – Curso de Sistemas de Informação, Departamento de Sistemas de Informação, Universidade Federal de Sergipe, Itabaiana, 2018.

## RESUMO

As ferramentas ROLAP representam uma parcela significativa do mercado de software voltado para análise e geração de consultas em ambientes de suporte à decisão. O conhecimento sobre a arquitetura dessas ferramentas é crucial para os profissionais que trabalham na área de inteligência voltada para os negócios. Esse conhecimento facilita o processo de implementação, manutenção, migração para outras tecnologias e aquisição de produtos. No entanto, as ferramentas existentes no mercado, sejam elas comerciais ou não, em função da sua complexidade, impedem que estudantes e profissionais consigam entender seus principais módulos e perceber que as diferentes ferramentas utilizam os mesmos princípios. Nesse trabalho, através da definição de um conjunto mínimo de requisitos necessários a uma solução ROLAP, foi desenvolvido um protótipo para auxiliar o ensino sobre características arquiteturais e módulos dessa classe de ferramenta. As primeiras avaliações do protótipo, realizadas através da utilização de exercícios de modelagem dimensional e da análise do seu código, mostraram indícios que ele pode ser um forte aliado ao ensino de ferramentas de suporte à decisão.

### Palavras-chave:

Ferramenta OLAP, *Data Warehouse*, Sistemas de Apoio à Decisão.

## **ABSTRACT**

ROLAP tools represent a significant part of the software market focused on analysis and query generation in decision support environments. Knowledge of the architecture of these tools is crucial for professionals working in the area of business-minded intelligence. This knowledge facilitates the process of implementation, maintenance, migration to other technologies and procurement of products. However, the existing tools in the market, whether commercial or not, due to their complexity, prevent students and professionals from being able to understand their main modules and realize that the different tools use the same principles. In this work, through the definition of a minimum set of requirements required for a ROLAP solution, a prototype was developed to assist in teaching about architectural features and modules of this class of tool. The first evaluations of the prototype, carried out through the use of dimensional modeling exercises and the analysis of its code, have shown that it can be a strong ally in the teaching of decision support tools.

### **Keywords:**

OLAP Tool, Data Warehouse, Decision Support Systems.



## LISTA DE FIGURAS

Figura 1 Teia de Aranha .....	16
Figura 2 Modelagem Dimensional .....	20
Figura 3 Cubo de Dados .....	21
Figura 4 OLAP .....	23
Figura 5 Arquitetura da Ferramenta de Administrador .....	27
Figura 6 Arquitetura da Ferramenta de Usuário .....	28
Figura 7 Diagrama de casos de uso da ferramenta de administrador .....	31
Figura 8 Diagrama de classes dos casos de uso das seções 4.1.3.1, 4.1.3.2 e 4.2.3.3 ....	40
Figura 9 Diagrama de classes do caso de uso manter fonte de dados .....	42
Figura 10 Diagrama de classes do caso de uso manter cubo de dados.....	43
Figura 11 Diagrama de classes do caso de uso manter fato .....	44
Figura 12 Diagrama de classes do caso de uso manter dimensão .....	45
Figura 13 Diagrama de classes do caso de uso manter tabela .....	45
Figura 14 Diagrama de classes do caso de uso manter atributo do fato .....	46
Figura 15 Diagrama de classes do caso de uso manter atributo da dimensão .....	47
Figura 16 Diagrama de classes do caso de uso manter relacionamento .....	48
Figura 17 Diagrama de casos de uso da ferramenta de usuário.....	50
Figura 18 Diagrama de classe do caso de uso abrir repositório .....	53
Figura 19 Diagrama de classe do caso de uso selecionar cubo .....	54
Figura 20 Diagrama de classe do caso de uso criar consulta.....	55
Figura 21 Esquema de dados da ferramenta .....	56
Figura 22 Esquema estrela utilizado.....	60
Figura 23 Tela Inicial da ferramenta .....	61
Figura 24 Tela de Criar repositório .....	62
Figura 25 Tela de Abrir Repositório .....	63
Figura 26 Tela de Criar Link para Repositório .....	64
Figura 27 Tela de Criar Fonte de Dados.....	65
Figura 28 Tela de Editar ou Remover Fonte de Dados .....	66
Figura 29 Tela de Criar Cubo de Dados .....	67
Figura 30 Tela de Editar ou Remover Cubo de Dados.....	68
Figura 31 Tela de adicionar e remover tabela, adicionar e remover fato e adicionar e remover dimensão.....	69
Figura 32 Caixa para adicionar um nome para o fato ou para a dimensão.....	69
Figura 33 Caixa para adicionar o atributo do fato .....	70
Figura 34 Caixa para adicionar o atributo da dimensão .....	71
Figura 35 Tela para adicionar e remover relacionamento .....	72
Figura 36 Tela inicial.....	72
Figura 37 Abrir repositório.....	73
Figura 38 Selecionar cubo de dados .....	74
Figura 39 Criar consulta 1 .....	75
Figura 40 Criar consulta 2 .....	75
Figura 41 Criar consulta 3 .....	76
Figura 42 Caixa de texto para cláusulas .....	76

## LISTA DE QUADROS

Quadro 1 Requisitos funcionais e não funcionais da ferramenta .....	30
Quadro 2 Caso de Uso Criar Repositório .....	33
Quadro 3 Caso de Uso Criar Link Para Repositório .....	33
Quadro 4 Abrir Repositório.....	33
Quadro 5 Criar fonte de dados.....	34
Quadro 6 Editar fonte de dados .....	34
Quadro 7 Remover fonte de dados .....	34
Quadro 8 Criar cubo de dados .....	35
Quadro 9 Editar cubo de dados.....	35
Quadro 10 Remover cubo de dados.....	36
Quadro 11 Adicionar fato .....	36
Quadro 12 Remover fato .....	36
Quadro 13 Adicionar dimensão.....	36
Quadro 14 Remover dimensão .....	37
Quadro 15 Adicionar tabela.....	37
Quadro 16 Remover tabela .....	37
Quadro 17 Adicionar atributo do fato.....	38
Quadro 18 Remover atributo do fato .....	38
Quadro 19 Alterar visibilidade .....	38
Quadro 20 Adicionar atributo da dimensão.....	39
Quadro 21 Remover atributo da dimensão .....	39
Quadro 22 Alterar visibilidade .....	39
Quadro 23 Adicionar relacionamento.....	39
Quadro 24 Remover relacionamento .....	40
Quadro 25 Requisitos funcionais e não funcionais da ferramenta .....	49
Quadro 26 Abrir repositório .....	51
Quadro 27 Selecionar cubo de dados .....	51
Quadro 28 Adicionar atributo.....	52
Quadro 29 Remover atributo .....	52
Quadro 30 Adicionar cláusula .....	52
Quadro 31 Remover cláusula .....	52

# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>11</b>
<b>1.1. Objetivos .....</b>	<b>12</b>
1.1.1. Geral .....	12
1.1.2. Específicos .....	12
<b>1.2. Relevância do trabalho .....</b>	<b>13</b>
<b>1.3. Metodologia .....</b>	<b>13</b>
<b>1.4. Estrutura do trabalho.....</b>	<b>14</b>
<b>2. REVISÃO BIBLIOGRÁFICA.....</b>	<b>15</b>
<b>2.1. Referencial Teórico.....</b>	<b>15</b>
2.1.1. Ambientes de Suporte À Decisão .....	15
2.1.2. Data Warehouse .....	17
2.1.3. Modelagem Dimensional.....	19
2.1.4. Ferramentas OLAP (Online Analytical Processing) .....	22
<b>2.2. Trabalhos relacionados no âmbito de Ambientes de Suporte à Decisão.....</b>	<b>25</b>
2.2.1. Neuro BI: Uma Ferramenta Web <i>open source</i> para o apoio à Tomada de Decisão .....	25
2.2.2. Uma Ferramenta em Software Livre para o Desenvolvimento de Sistemas de Suporte à Decisão .....	25
<b>3. ARQUITETURA PROPOSTA PARA A SOLUÇÃO.....</b>	<b>27</b>
<b>3.1. Ferramenta de Administrador (Módulo de configuração do repositório).....</b>	<b>27</b>
<b>3.2. Ferramenta de Usuário (Módulo de criação de consultas) .....</b>	<b>28</b>
<b>4. PROJETO DA FERRAMENTA .....</b>	<b>29</b>
<b>4.1. Ferramenta de Administrador (Módulo de configuração do repositório).....</b>	<b>29</b>
4.1.1. Requisitos funcionais e não funcionais.....	29
4.1.2. Casos de Uso.....	30
4.2.2.1. Casos de Uso – Nível de Usuário .....	30
4.2.2.2. Casos de Uso – Nível de Sistema .....	32
4.1.3. Diagramas de Classe .....	40
4.2.3.1. Caso de Uso Criar Repositório.....	40
4.2.3.2. Caso de Uso Criar Link para Repositório .....	41
4.2.3.3. Caso de Uso Abrir Repositório .....	41
4.2.3.4. Caso Manter Fonte de Dados (criar, editar e remover) .....	41
4.2.3.5. Caso Manter Cubo de Dados (criar, editar e remover) .....	42
4.2.3.6. Caso de Uso Manter fato (adicionar e remover).....	43
4.2.3.7. Caso de Uso Manter dimensão (adicionar e remover) .....	44
4.2.3.8. Caso de Uso Manter tabela (adicionar e remover) .....	45
4.2.3.9. Caso de Uso Manter Atributo do Fato .....	46
4.2.3.10. Caso de Uso Manter Atributo da Dimensão .....	47
4.2.3.11. Caso de Uso Manter Relacionamento.....	48
<b>4.2. Ferramenta de Usuário (Módulo de criação de consultas).....</b>	<b>49</b>
4.2.1. Requisitos funcionais e não funcionais.....	49
4.2.2. Casos de Uso.....	50
4.2.2.1. Casos de Uso – Nível de Usuário .....	50
4.2.2.2. Casos de Uso – Nível de Sistema .....	50
4.2.3. Diagramas de Classe .....	52
4.2.3.1. Caso de Uso Abrir Repositório .....	53
4.2.3.2. Caso de Uso Selecionar Cubo de Dados .....	53

4.2.3.3. Caso de Uso Criar consulta (adicionar atributos, remover atributos, adicionar cláusula, remover cláusula) .....	54
<b>4.4. Modelagem de Dados.....</b>	<b>56</b>
4.4.1. Componentes do modelo de dados.....	56
<b>5. FERRAMENTA .....</b>	<b>60</b>
5.1. Ferramenta de Administrador (Módulo de configuração do repositório).....	60
5.2. Ferramenta de Usuário (Módulo de criação de consultas) .....	72
<b>6. CONCLUSÃO E TRABALHOS FUTUROS .....</b>	<b>77</b>
<b>REFERÊNCIAS.....</b>	<b>79</b>

## 1. INTRODUÇÃO

Um sistema de apoio à decisão é um conjunto de procedimentos baseados em modelos para o processamento de dados e julgamentos para ajudar o seu utilizador a tomar decisões (LITTLE, 1970). Até a década de 80 as informações eram analisadas e geradas diretamente do ambiente OLTP (*On-line Transaction Processing*), no qual predominam sistemas operacionais de registro que capturam as transações do negócio (KIMBALL, 2002). Embora a maior especialidade do OLTP seja a inserção dos dados no banco de dados, ele não é eficiente em análises de dados gerenciais, pois geralmente demoram ou falham na recuperação ou apresentação das informações. É fato que o negócio começa a sofrer imediatamente quando o tempo de resposta torna-se ruim no OLTP (INMON, 2005). A solução encontrada para a melhoria na extração de informações dos dados, foi a criação do ambiente de *Data Warehouse* (DW).

O ambiente de *Data Warehouse* representa um ambiente separado do ambiente operacional para suporte à tomada de decisão. Por estar separado, esse novo ambiente não interfere nas operações diárias dos sistemas OLTP e pode apresentar estruturas de dados específicas que auxiliem a extração de informações. No centro desse ambiente, encontra-se sua peça fundamental, o *Data Warehouse*, um banco de dados histórico, integrado, concebido a partir de dados dos ambientes legados e externos, voltado para a tomada de decisões.

Segundo Inmon (2005), *Data Warehouse* é uma coleção de dados orientado a assuntos, integrado, tempo-variante, e não-volátil, para suporte ao gerenciamento dos processos de tomada de decisão. Ele define cada conceito desses da seguinte forma: a) Orientado a assuntos – Os esquemas de dados de um DW geralmente dizem respeito a um único processo de negócio, ao contrário os esquemas operacionais que tratam de vários processos. b) Integrados – Os dados são carregados para o *Data Warehouse* a partir de uma variedade de origens e reunidos em um todo coerente. c) Tempo-variante – Todo esquema em um DW sempre apresenta um componente. d) Não-voláteis – Os dados sempre são adicionados, nunca removidos ou modificados.

O ambiente de *Data Warehouse*, geralmente, utiliza uma técnica de modelagem de dados no nível lógico conhecida como modelagem dimensional. Os modelos dimensionais são compreensíveis, previsíveis e ampliáveis, pois se mantém fiel à

simplicidade tendo uma perspectiva voltada para as necessidades analíticas da empresa (KIMBALL, 2002).

Com o surgimento do *Data Warehouse*, foi percebida a necessidade de criar ferramentas que auxiliassem na criação de relatórios consistentes e que tivesse capacidade de análise de dados maior do que já se analisava no momento. Foi a partir desse momento que surgiram as ferramentas OLAP (*Online Analytical Processing*). Kimball define OLAP como a atividade de consulta e apresentação de dados textuais e numéricos em um DW. É a atividade de análise dos dados do ambiente de *Business Intelligence* para a extração de significado. Esse significado é a base para o processo de tomada de decisão.

Independente da arquitetura utilizada, seja ela ROLAP (*Relational OLAP*), MOLAP (*Multidimensional OLAP*) ou HOLAP (*Hybrid OLAP*), todas as ferramentas OLAP compartilham, no mínimo, conceitos e módulos que são mandatórios. Dentre esses conceitos e módulos podemos destacar: a) O Servidor olap; b) O repositório de metadados; c) os esquemas lógicos; d) os esquemas semânticos; e) os esquema de apresentação; e) o mapeamento entre os esquemas; f) as fontes de dados; g) as características dos fatos; h) as hierarquias; e i) os agregados.

## **1.1. Objetivos**

### **1.1.1. Geral**

O objetivo desse trabalho é criar um protótipo de ferramenta ROLAP para auxiliar o ensino das características e arquitetura de ferramentas OLAP.

### **1.1.2. Específicos**

- Definir um conjunto mínimo de características de uma arquitetura ROLAP para compor a solução proposta;
- Especificar e Projetar o repositório de metadados para a arquitetura ROLAP;
- Especificar, Projetar e Construir a ferramenta para criação do repositório de metadados e criação das camadas semântica e de mapeamento;
- Especificar, Projetar e Construir a ferramenta de análise de dados e geração de consultas;

- Realizar testes com Exercícios de Modelagem da disciplina de Sistemas de Apoio à Decisão;

## 1.2. Relevância do trabalho

O conhecimento de ferramentas OLAP é crucial para os profissionais que trabalham em ambientes de suporte à decisão. Esse conhecimento facilita o processo de implementação, manutenção, migração para outras tecnologias e aquisição de ferramentas. As ferramentas existentes no mercado, sejam elas comerciais ou não, em função da sua complexidade, impedem que estudantes e profissionais consigam perceber e entender os principais módulos que compõe suas arquiteturas. Essa realidade foi percebida por professores do Departamento de Sistemas de Informação com especialidade na área, atuando desde a década de 90, que lecionam as disciplinas de Sistemas de Suporte à Decisão na graduação e pós-graduação, e que já participaram de projetos de ambiente de *Data Warehouse* no setor público e privado.

Em função dessa realidade, o objetivo desse trabalho é criar um protótipo de ferramenta ROLAP, contemplando a arquitetura e os módulos mínimos, citados anteriormente, de uma ferramenta comercial, como MicroStrategy (<https://www.microstrategy.com/br>), Power BI (<https://powerbi.microsoft.com/pt-br/>), Tableau (<https://www.tableau.com/>), Oracle BI (<https://www.oracle.com/br/solutions/business-analytics/index.html>) e Pentaho (<http://www.pentaho.com/>), com o intuito de auxiliar o ensino das características e arquitetura de ferramentas OLAP para estudantes da área de Sistemas de Informação e Ciência da Computação.

## 1.3. Metodologia

Inicialmente foi realizada a revisão bibliográfica sobre Ambiente de Suporte à Decisão, *Data Warehouse*, Modelagem Dimensional e Ferramentas OLAP.

Após a revisão bibliográfica, foram elicitados os requisitos, tanto funcionais, quanto não funcionais, descrevendo todos os aspectos significativos da arquitetura proposta. Essa atividade de elicitação teve como principal insumo as necessidades apontadas pelo professor da disciplina de Sistemas de Suporte à Decisão.

Após a elicitação e análise dos requisitos, foram projetados e construídos, de maneira incremental, os módulos que compõem a solução: o repositório de metadados, as ferramentas administrativas e a ferramenta de análise e consulta.

#### **1.4. Estrutura do trabalho**

O restante do trabalho está estruturado como segue. No capítulo 2 são apresentados alguns trabalhos encontrados relacionados com o tema do trabalho e uma revisão bibliográfica. No capítulo 3 é apresentada a arquitetura proposta para a solução. No capítulo 4, são descritas as principais atividades realizadas durante o projeto da ferramenta. O capítulo 5 apresenta as funcionalidades da ferramenta através do seu uso com um esquema de dados para sistemas de *helpdesk*. No capítulo 6 são apresentadas as conclusões e as propostas de trabalhos futuros.



## **2. REVISÃO BIBLIOGRÁFICA**

### **2.1. Referencial Teórico**

Nesta seção são apresentados os principais conceitos das áreas de Sistemas de Suporte à Decisão, *Data Warehouse*, Modelagem Dimensional e Ferramentas OLAP necessários para o entendimento da solução proposta e das discussões sobre a arquitetura e projeto nos capítulos seguintes.

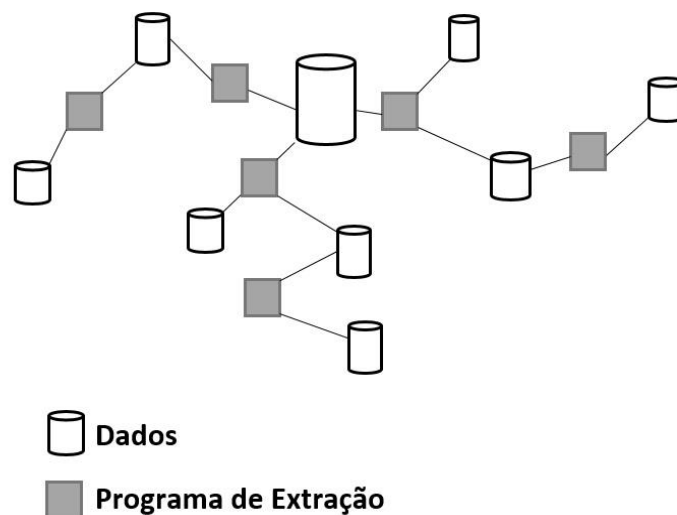
#### **2.1.1. Ambientes de Suporte À Decisão**

Em Inmon (2005) é dito que os sistemas de suporte à decisão evoluíram juntamente com a tecnologia da informação. As aplicações desse tipo faziam uso de arquivos mestres para emissão de relatórios. Nessa época, anos 60, eles eram armazenados em fitas magnéticas, que apesar da grande quantidade de dados que elas podiam guardar, todos eles precisavam ser acessados em sequência. Normalmente acessava-se todos os dados para encontrar o que realmente se procurava, deixando todo o processo lento e com tempo indeterminado de conclusão. Com isso, na década seguinte, em 1970, surgiu uma nova tecnologia, que foi o disco rígido, que permitia um acesso direto à informação, já não sendo mais necessário percorrer todos os registros em sequência, diminuindo o tempo de forma considerável. Acompanhando o surgimento do disco rígido também surgiram os Sistemas Gerenciadores de Banco de Dados (SGBD). Esse tinha como função facilitar o armazenamento e o acesso aos dados contidos no disco rígido. Além disso, o SGBD trouxe melhor entendimento sobre o funcionamento de bancos de dados, fazendo com o que os SGBDs fossem usados como fontes exclusivas de processamento de dados, que logo foi melhorado no ambiente OLTP (*Online Transaction Processing*). Com essa evolução, os computadores passaram a ser usados por todos os tipos de usuários e não só por profissionais da computação, fazendo com que fosse possível ter os sistemas que temos nos dias atuais.

Em 1980, a tecnologia continuou a evoluir, trazendo computadores mais potentes e novas linguagens de programação, fazendo com que usuários comuns conseguissem ter controle sobre sistemas e dados, e evidenciando que muito mais poderia ser feito com os dados de uma organização. Nesse momento surgiu o que conhecemos hoje por sistemas de apoio à decisão, que é um sistema que auxilia na tomada de decisões em nível de gerência. Surgiam nesse cenário, os programas de extração, que serviam para extrair

informações dos sistemas OLTP cada que vez maiores. Os dados eram extraídos para outro lugar que não interferiam no processamento online enquanto a análise era feita.

Com vários extratos de diversos lugares, até mesmo extratos dos extratos existentes, os dados se assemelhavam a uma “teia de aranha”. Esse ambiente era muito comum nas organizações, que quanto maiores, mais problemas tinham, por causa de tal organização das informações. Na figura 01 podemos observar o relacionamento das várias bases e dados com os programas de extração.



**Figura 1 Teia de Aranha**

Esse tipo de arquitetura possui alguns problemas. Um deles é que pode-se extrair a informação de um conjunto de dados gerando ambiguidade na geração de relatórios, uma vez que consultas em lugares diferentes podem trazer diferentes informações sobre um mesmo assunto. Outro problema acontece com a produtividade, já que quando se faz necessário que seja realizado algum relatório sobre algum conjunto de dados, a grande quantidade deles, exige que se percorra por todas as fontes, com o objetivo de descobrir onde está a informação desejada. Além dessas falhas, ainda existe a incapacidade de converter os dados em informações concisas. Para satisfazer a uma consulta o analista deve se deparar com dados não integrados vindos de várias fontes diferentes. Uma outra falha nesse ambiente é a falta de histórico de dados para análise, pois essa arquitetura é projetada para atender o momento atual do negócio. Com tantas falhas graves, foi percebido que esse modo de armazenar os dados não era satisfatório às necessidades futuras do negócio. Então para melhorar o desempenho e qualidade das informações, foi dado origem a um novo conceito chamado de *Data Warehouse*.

Com esse novo conceito, o *Data Warehouse*, os dados de um ambiente operacional diferem dos dados que ficam armazenados no ambiente de suporte à decisão. Os dados do ambiente operacional podem ser atualizados e auxiliam nas operações online que são realizadas durante o uso do sistema. Já os dados de um ambiente de suporte à decisão são refinados, categorizados por assuntos e sempre são acompanhados por valores de tempo. São acessados por conjuntos e sempre fazem controle de atualização, não permitindo atualizações diretas como no ambiente operacional. Esses dados auxiliam na gestão, ajudando na resolução de problemas ou de supostas situações futuras.

### **2.1.2. Data Warehouse**

De acordo com Inmon (2005) o *Data Warehouse* é uma coleção de dados orientados por assunto, integrado, não volátil e variantes no tempo.

Orientado por assunto significa que os dados são separados por categorias a depender do negócio que está sendo usado, pois cada organização tem diferentes conjuntos de assuntos. Dados integrados são dados que vêm de diversas fontes quando colocados no *Data Warehouse*, tem uma forma coerente quando se olha o todo. Em outras palavras os dados quando são pegos no ambiente operacional recebem o tratamento para povoar o *Data Warehouse* e não sofrem nenhum tipo de distorção quando são vistos em conjunto. Um exemplo disso é o uso de padrões em nomenclaturas em atributos ou estruturas de chaves. A não volatilidade é outra característica do *Data Warehouse*. É dito que no ambiente operacional os dados são acessados e atualizados de forma regular, usando um registro por vez. Já no *Data Warehouse* os dados são diferentes, pois sempre são acessados em massa criando novos registros a cada mudança, gerando e mantendo assim um histórico de dados. Por último vemos mais uma característica do *Data Warehouse*, que é a variância no tempo. Com ela vemos que cada registro do datawarehouse é gravado em algum instante no tempo. Deixando claro que o *Data Warehouse* possui muito mais histórico do que outros ambientes.

O *Data Warehouse* é composto por vários *Data Marts*. *Data Marts* são subconjuntos de dados contidos em um *Data Warehouse* (COLAÇO JR, 2004). Geralmente nos *Data Marts* são armazenados dados sobre um processo específico da empresa. Vários *Data Marts* juntos podem formar o DW completo.

Seja qual for a definição, o objetivo final do *Data Warehouse* é integrar dados empresariais de toda a empresa em um único repositório a partir do qual os usuários podem facilmente executar consultas, produzir relatórios e realizar análises (CONNOLLY; BERG, 2015).

Em (CONNOLLY; BERG, 2015) vemos que o *Data Warehouse* traz vários benefícios a um negócio. Um deles é que quando são feitos grandes investimentos na criação de um *Data Warehouse* ele traz retornos muito mais positivos, já que outra vantagem dele é a competitividade no mercado, pois ele permite melhor análise dos dados na tomada de decisão influenciando na produtividade da organização, transformando dados em informações de suma importância para decisões futuras ou análise de situações.

O *Data Warehouse* difere do ambiente OLTP em vários quesitos. O ambiente OLTP é projetado para realizar análise operacional do negócio e online, já o DW é usado para realizar processos de análises que são armazenados em determinados ciclo de tempo. O ambiente OLTP se preocupa principalmente com a inserção, atualização ou eliminação de dados, já o DW trabalha com a extração de informações que os dados podem oferecer mantendo históricos detalhados. Normalmente uma empresa possui um OLTP para cada processo do negócio da empresa, onde só é necessário um DW para operar sobre eles. É nítido que além das diferenças nas características, os dois ambientes trabalham juntos, basicamente o OLTP coletando os dados que serão analisados pelo DW.

Em Kimball (2002) a arquitetura de um DW é basicamente definida pelos seguintes componentes: O ambiente operacional, área de testes de dados (onde ocorre o *Extract Transform Load* (ETL)), o DW em si e as ferramentas de análises.

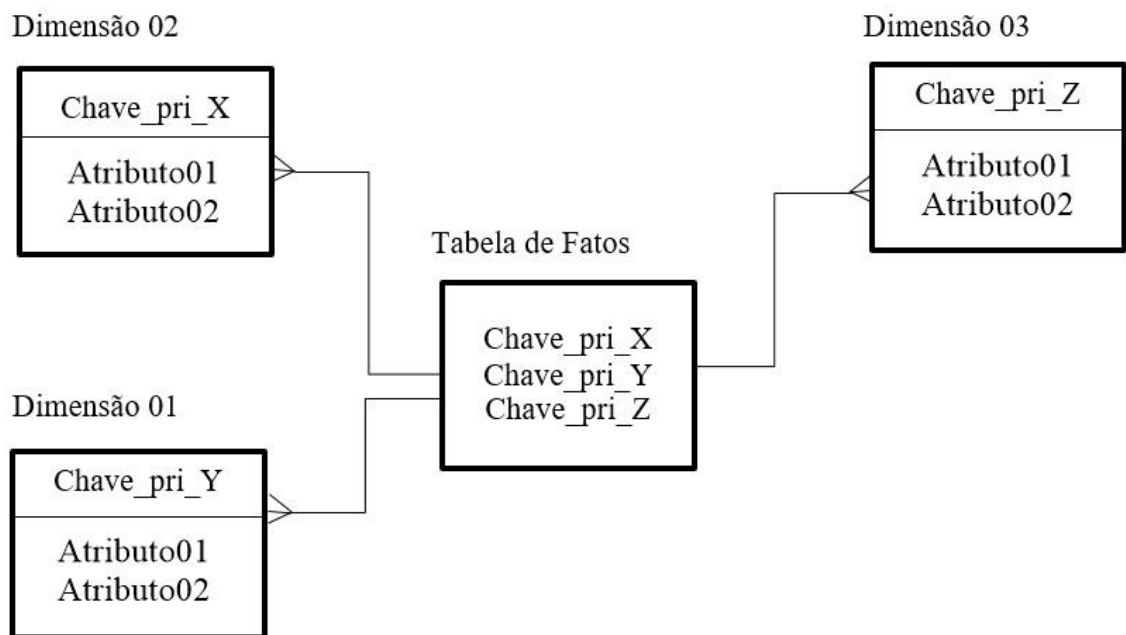
O ambiente operacional é responsável por obter os dados do negócio da organização através das transações. Ele é pensado como uma fonte separada do Data Warehouse pois o seu foco é alto desempenho e acessibilidade fácil aos dados. Os dados são extraídos do ambiente operacional e passam por um processo de transformação e de limpeza. São verificados os padrões de formatos ou de nomes de atributos, correções ortográficas, divisão de tabelas e até unificação de algumas delas, realizando a combinação de dados que chegam de fontes distintas. Não é permitido realizar consulta de dados quando os mesmos estão passando pelos processos dessa área. Quando os dados são transformados eles já saem prontos para serem inseridos no DW em forma de tabelas dimensionais. Quando jogados no DW, os dados já estão prontos para serem consultados diretamente pelos usuários e por ferramentas de análises. Tais ferramentas acessam os

dados diretamente no DW e facilitam na apresentação das informações da organização, atendendo aos usuários de forma que não exijam que eles criem consultas diretas.

### **2.1.3. Modelagem Dimensional**

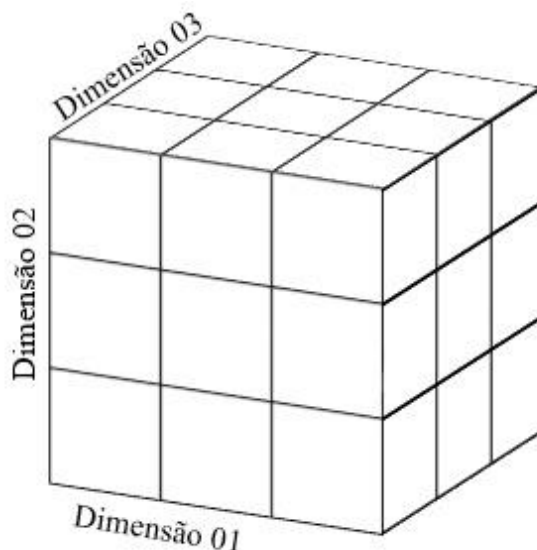
De acordo com Kimball (1997, 2002) a modelagem dimensional é o nome de uma técnica de projeto lógico usada frequentemente para *Data Warehouse* e viável para bancos de dados projetados para suportar consultas de usuários finais no mesmo e que buscam mostrar os dados de forma intuitiva e com alto desempenho no acesso deles. O modelo dimensional é composto por tabelas de fatos e tabela de dimensões. As dimensões são divididas em alguns tipos, sendo os principais, o tipo 1, tipo 2 e o tipo 3. No tipo 1 o valor é simplesmente substituído pelo novo valor, normalmente usado quando o histórico do campo não será importante para alguma decisão futura. O tipo 2 cria novos registros quando algum valor é alterado e isso fica registrado em intervalos de tempo com atributos de tempo. E no tipo 3 se criam atributos que registram os valores iniciais e os atuais, perdendo completamente os valores que ficam entre os dois.

Geralmente uma linha da tabela de fatos representa a principal transação do negócio. Essa tabela traz chaves primárias de várias outras tabelas que são conhecidas com tabelas de dimensões. Cada tabela dessas possui uma chave primária que compõe a tabela de fatos. Toda essa estrutura também é conhecida como esquema estrela. A tabela de fatos traz indicadores aditivos, semi-aditivos e não aditivos. Aditivos são atributos que podem ser somados no decorrer das dimensões, como unidades de algum produto. Semi-aditivos faz medições que não fazem sentido serem somadas como temperatura ou velocidade. E não-aditivos, que não podem se somar de forma alguma, como número do cliente, por exemplo.



**Figura 2 Modelagem Dimensional**

Esse modelo, mostrado na figura 02, traz algumas vantagens para ser usado no *Data Warehouse*. Primeiramente, esse modelo tem uma estrutura bem esperada, algo que facilita na geração de relatórios e na criação de ferramentas de consulta para melhor usabilidade ao usuário final, já que o esquema estrela suporta mudanças que não estejam previstas. Outra vantagem é que essa estrutura é bem flexível e favorável a extensões, podendo sofrer alterações ou adições nas tabelas, acrescentando linhas ou até mesmo alterando tabelas já existentes. Novos fatos podem ser adicionados, desde que sejam coerentes com a granularidade da tabela de fatos já existente. Com essas características, não se faz necessário que ferramentas de consulta precisem ser reformuladas para se adaptar às mudanças. Através das principais dimensões do negócio, cubos de dados com interseções dos dados das mesmas são criados, como pode-se ver abaixo.



**Figura 3 Cubo de Dados**

Em (CONNOLLY; BERG, 2015) podemos ver que o modelo dimensional e o modelo na terceira forma normal (3FN) possuem algumas diferenças, a primeira delas é que o modelo dimensional é normalmente usado para projetar *Data Warehouse*, já os modelos na terceira forma normal (3FN) são usados para o projeto de sistemas OLTP. O Modelo na 3FN diminui a redundância dos dados e isso é extramamente vantajoso para as transações que são relizadas no OLTP, porém não são suficientes para suprir as consultas maciças do usuários finais em um ambiente de surpote à decisão.

Também é visto em Kimball (1997, 2002) que apesar de bastante utilizada, a modelagem dimensional ainda possui alguns mitos. O primeiro dos mito é "A implementação de um modelo dimensional cria um projeto de difícil adaptação". Esse mito se originou quando tabelas de fatos foram prematuramente agregadas.

O segundo mito diz que "Ninguém entende modelagem dimensional". Esse mito é um dos mais inaceitáveis, pois esse modelo é visualmente fácil de compreender e normalmente não é preciso que o criador do modelo dimensional de algum negócio específico esteja ao lado para fazer explicações sobre o funcionamento.

O terceiro mito afirma que "Os modelos dimensionais só funcionam com bancos de dados de varejo". Esse é um dos mais fortes na modelagem dimensional, devido ao grande histórico, mas nos dias de hoje, isso mudou e o modelo pode ser usado em qualquer tipo de negócio.

E por fim, o quarto mito. Ele diz que "Floco de neve é uma alternativa à modelagem dimensional". Floco de neve (*snowflake*) é uma variação do esquema estrela em que as dimensões foram normalizadas. Isso, segundo Kimball (1997, 2002), não é

aceitável pois pode trazer muitos problemas ligados ao desempenho das consultas e ao tratamento de histórico.

É necessário seguir alguns passos para realizar uma boa modelagem dimensional. Primeiramente é necessário definir o funcionamento do negócio em que o modelo vai operar. Depois disso deve se definir a granularidade, que é o nível de detalhe que deve estar disponível no modelo. Com isso em mãos inicia a criação das dimensões. Normalmente são feitas algumas perguntas sobre o negócio (O que? Como? Quando? Quanto? Onde? E Outras). Por fim a identificação dos fatos na tabela de fatos. Essas escolhas são diretamente apoiadas pela granularidade.

A modelagem dimensional é a melhor técnica para se projetar bancos de dados que serão entregues a usuários, oferecendo um maior desempenho nas consultas e maior flexibilidade em relação às perspectivas desejadas pelos usuário.

#### **2.1.4. Ferramentas OLAP (Online Analytical Processing)**

OLAP (*Online Analytical Processing*) é a síntese, análise e consolidação dinâmica de grandes volumes de dados multidimensionais (CONNOLLY; BERG, 2015).

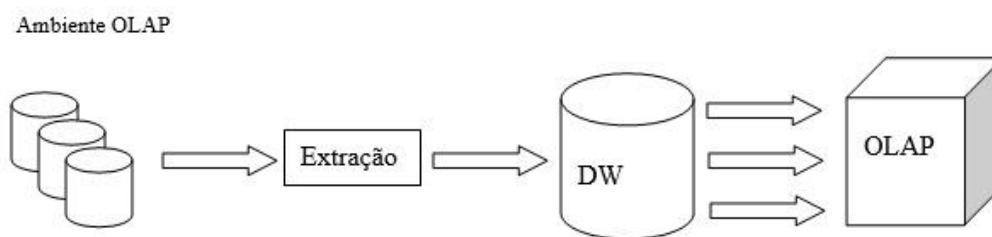
Em (COLAÇO JR, 2004) é dito que ferramentas OLAP possuem técnicas capazes de tratar as informações em um *Data Warehouse*. A característica de maior importância delas é a visão multidimensional dos dados.

Porém além dela, outras características também tem a sua importância. Em (CONNOLLY; BERG, 2015) podemos ver algumas delas. A capacidade de possuir uma visão multidimensionais dos dados. Com isso é possível coletar dados com mais flexibilidade. Possivelmente todas as dimensões são trazidas de forma igual, permitindo o melhor entendimento visual do negócio. O sistema OLAP também deve fornecer capacidade de realizar cálculos de níveis mais complexos, como cálculo de médias e crescimento percentual. E por fim a chave de aplicações de análise, os intervalos de tempos, já que normalmente dados são comparados em relação ao tempos, por exemplo, a comparação de dados do mês passado com o mês atual.

Um requisito essencial de todas as aplicações OLAP é a capacidade de fornecer informações aos usuários (CONNOLLY; BERG, 2015). Um bom servidor OLAP possui características como a carga de grandes massas de dados, agregação de dados,



organização de dados através de hierarquias, e análise de dados em intervalos de tempo. Na figura 04 podemos ver onde ferramentas OLAP se encaixam na estrutura do DW.



**Figura 4 OLAP**

De acordo com (COLAÇO JR, 2004), ferramentas OLAP possuem uma característica chamada de *Slice and Dice*. Sua função é manipular as dimensões a fim de facilitar a compreensão do usuário. Ela possui quatro operações: *Ranging*, *Drilling*, *Rotation* e *Ranking*.

O *Ranging* altera o resultado das consultas, dando novas posições ou retirando as que estão em foco. Sendo que tudo isso é informado pelo usuário. Já o *Drilling* permite que o analista mude a análise depois de escolher o que vai analisar, sendo que as informações podem se encontrar em níveis diferentes. Para essa mudança existem três operações: *Drill Down*, *Drill Up* e *Drill Across*. O *Drill Down* navega na hierarquia em direção ao menor detalhamento dos dados a fim de obter informações mais precisas. Com o *Drill Up* é feito o inverso do *Drill Down* indo em direção a informações mais gerais. E no *Drill Across* se permite que pule de níveis na mesma dimensão. Na operação *Rotation* é permitido que o analista mude a forma como ele quer olhar para as informações sem acrescentar ou remover nada. E por fim, o *Ranking*. Esse realiza filtros nas informações que se quer obter, fazendo classificações que ficam a critério do analista.

Em 1993, Codd formulou 12 regras que caracterizam os requisitos de uma ferramenta OLAP:

1. Visão multidimensional: As ferramentas OLAP devem permitir que os usuários tenham uma visão multidimensional dos fatos de uma empresa a fim de facilitar o uso da ferramenta.
2. Transparência: O banco, as fontes de dados, a arquitetura. Tudo deve ser transparente a fim de preservar a produtividade e a usabilidade do usuário.

3. Acessibilidade: A ferramenta OLAP deve permitir o acesso dos dados que serão usados para a realização de alguma análise.
4. Desempenho: Independente do número de dimensões ou do tamanho do banco de dados, o usuário não pode perceber nenhuma falha no desempenho das consultas.
5. Arquitetura cliente-servidor: O sistema OLAP deve possuir um ambiente cliente-servidor.
6. Dimensionamento genérico: Deve ser capaz de tratar informações em qualquer quantidade de dimensões.
7. Manipulação dinâmica da matriz esparsa: A ferramenta deve se adaptar ao esquema físico a fim de melhorar o desempenho e rapidez no acesso aos dados.
8. Suporte multiusuário: Deve ser possível que vários usuários usem a ferramenta de forma paralela nos dados de uma organização.
9. Operações transversais sem restrições: Deve reconhecer as hierarquias existentes nas dimensões.
10. Manipulação de dados intuitiva: As manipulações devem ser feitas de forma intuitiva, clicando, arrastando ou apontando, com o propósito de realizar alguma ação.
11. Relatórios flexíveis: Os relatórios devem ser organizados em linhas e colunas de uma forma que facilite o entendimento do usuário.
12. Dimensões ilimitadas e níveis de agregação: Um negócio pode possuir várias dimensões, onde cada uma possui suas quantidades de hierarquias. Com isso não deve haver restrições em seus números.

Além dessas 12 regras de Codd, foram acrescentadas mais 9 pelo Garther Group, sendo elas:

1. Dados Arrays múltiplos;
2. OLAP joins;
3. Ferramentas para gerenciar as bases de dados;
4. Armazenar objetos;
5. Seleção de subconjuntos;
6. Detalhe drill-down em nível de linha
7. Suporte a dados locais;
8. Refresh incremental das bases de dados;

## 9. Interface SQL.

Existem vários tipos de arquiteturas OLAP, sendo algumas delas, a HOLAP (*Hybrid OLAP*), MOLAP (*Multidimensional OLAP*), ROLAP (*Relational OLAP*). Nesse trabalho será usada apenas a arquitetura ROLAP. Os sistemas ROLAP permitem análise multidimensional dos dados que estão armazenados em uma base de dados relacional, podendo ser feito todo o processamento no servidor da base de dados e depois gerando os comandos SQL e as tabelas temporárias (COLAÇO JR, 2004).

## 2.2. Trabalhos relacionados no âmbito de Ambientes de Suporte à Decisão

Nesta seção apresentaremos algumas propostas de ferramentas voltadas à ambientes de suporte à decisão. Não foram encontrados até o momento trabalhos que relatem ferramentas que tenham como abordagem a implementação de uma arquitetura ROLAP voltada para o ensino.

### 2.2.1. Neuro BI: Uma Ferramenta Web *open source* para o apoio à Tomada de Decisão

Em (FERNANDES; KANTORSKI, 2008) é proposta uma ferramenta que faz o uso dos conceitos de *Business Intelligente*, *Data Warehouse* e Arquiteturas OLAP para gerar funcionalidades para interação com o usuário. Na ferramenta é feito o uso de um servidor Web, utilizando como gerenciador de base de dados o PostgreSQL. Além dos assuntos citados acima, são usados três frameworks por terem disponibilidade de recursos na linguagem Java. Sendo eles: *Sitemesh*, que faz o layout das páginas, *Hibernate*, que manipula o banco de dados, e *Spring*, que é a lógica do negócio em si. O objetivo da ferramenta Neuro BI é realizar abstração de todos os conceitos usados sem conhecimento da linguagem MDX. Mas caso tenha conhecimento, o usuário pode alterar ou criar novas consultas.

### 2.2.2. Uma Ferramenta em Software Livre para o Desenvolvimento de Sistemas de Suporte à Decisão

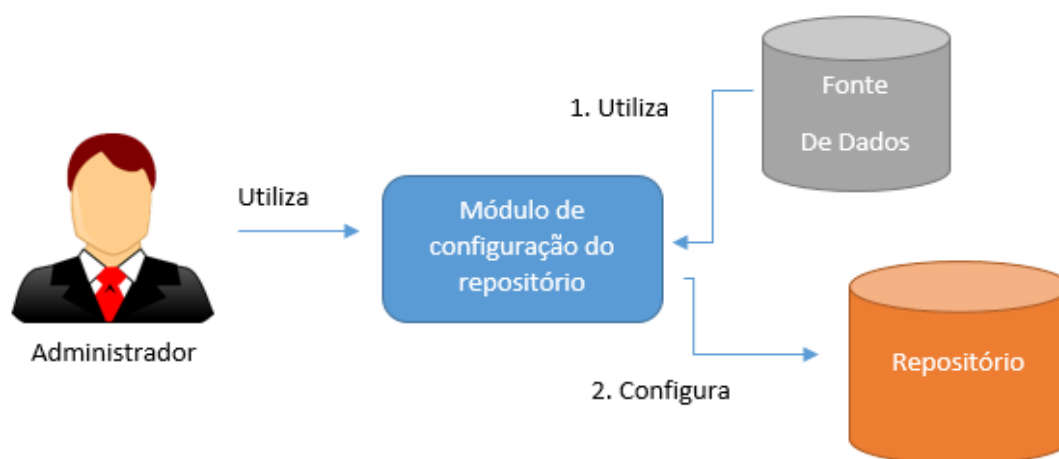
No trabalho de (COLONESE; MANHÃES; GONZÁLEZ; GALANTE; CARVALHO; TANAKA, 2008) é proposta uma ferramenta livre e de código aberto para suporte à decisão

que integra tecnologias OLAP e SIG (Sistemas de Informação Geográfica). A ferramenta se chama PostGeoOlap e tem como objetivo permitir a construção de sistemas de suporte à decisão, onde seja possível tratar aspectos analíticos e geográficos. A ferramenta realiza consultas em um *Data Warehouse* que contém dimensões com atributos geográficos. A PostGeoOlap é uma ferramenta para a criação de soluções OLAP espaciais sobre o SGBD PostgreSQL e o PostGIS. O nome PostGeoOlap tem origem na integração entre funcionalidades geográficas, tecnologia OLAP e PostgreSQL. Para a construção da ferramenta, adotou-se o paradigma de Software Livre, assim, todas as APIs, frameworks e softwares de persistência de dados utilizados neste projeto são *open source*. Também foi adotado o ROLAP como modelo, assim, ambas as pesquisas, geográficas ou analíticas convencionais, serão processadas e respondidas pelo SGBD PostgreSQL e todos os dados, desde o nível base até as agregações, serão mantidos no modelo relacional. A ferramenta consegue criar esquemas, criar cubos selecionando os itens, operações e os fatos, adicionar dimensões e realizar a análise dos dados.

### 3. ARQUITETURA PROPOSTA PARA A SOLUÇÃO

Este capítulo apresenta a arquitetura da solução proposta. São apresentadas as arquiteturas das duas principais ferramentas. A ferramenta de Administrador, responsável pela criação e gerenciamento do repositório e cubos de dados; e a ferramenta de Usuário, responsável pela parte de visualização dos cubos e criação de consultas.

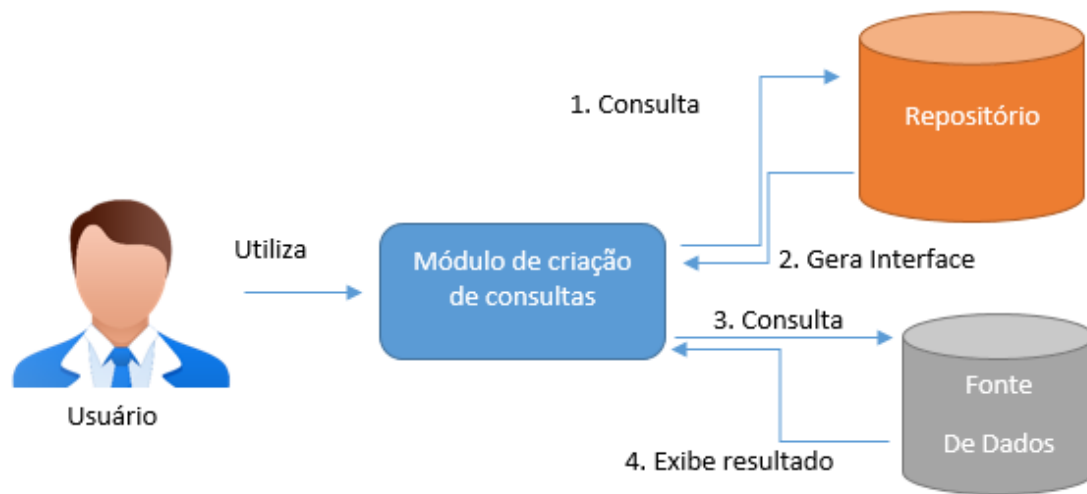
#### 3.1.Ferramenta de Administrador (Módulo de configuração do repositório)



**Figura 5 Arquitetura da Ferramenta de Administrador**

Nesta ferramenta, como visto na figura 5, o administrador do sistema realiza a configuração do repositório de dados que será usado pelo usuário comum para a criação de consultas. Para isso, administrador pode criar o repositório ou acessar um que já exista. Com o repositório acessado, tem início o processo de criação de um cubo de dados, e para isso é necessário criar um link para acessar uma fonte de dados que já existe na máquina e, através desses dados a configuração em si se inicia, o administrador enxerga todas as tabelas daquela fonte de dados e identifica qual é a tabela de fato e quais são as tabelas de dimensões, tendo como alternativa nomear cada um com um nome que seja mais conveniente. Depois disso é necessário que se identifique quais atributos serão disponibilizados para a utilização do usuário para a criação de consultas, podendo também ocultá-los. E por fim, o administrador utiliza esse módulo para identificar como a tabela de fatos se relaciona com as dimensões, a fim de proporcionar a formação da consulta no outro módulo. Além disso, é bom ressaltar que é possível excluir, ou alterar tudo depois de já criado.

### 3.2.Ferramenta de Usuário (Módulo de criação de consultas)



**Figura 6 Arquitetura da Ferramenta de Usuário**

Como visto na figura 6, esse módulo tem a função de exibir os dados do cubo da maneira como foi configurado na ferramenta anterior. Através de uma interface o usuário consulta o repositório, gerando um *query* que é enviada para a fonte de dados, retornando os resultados dela. Nesse modelo são mostrados quais os fatos e as dimensões disponíveis para utilização, bastando apenas um clique sobre o que foi escolhido para o mesmo ser inserido numa *query* que já é um padrão para consultas nesse modelo. Essa consulta além de ser exibida na tela, também é executada trazendo ao usuário os resultados.

## 4. PROJETO DA FERRAMENTA

Após a revisão bibliográfica, iniciou-se o projeto da ferramenta, no qual foram analisados e definidos quais seriam seus requisitos funcionais e não funcionais. Além disso, foram modelados o digrama de casos de uso e o projeto lógico de dados.

### 4.1. Ferramenta de Administrador (Módulo de configuração do repositório)

#### 4.1.1. Requisitos funcionais e não funcionais

Os requisitos funcionais da ferramenta, resumidos no Quadro 1, são: 1) Criar repositório. Esse requisito permite que sejam criados repositório de dados localmente ou em qualquer máquina da mesma rede; 2) Criar link para repositório. É permitido criar links de conexão para acessar repositórios locais ou que estão em uma máquina da mesma rede; 3) Abrir repositório. Esse requisito permite que um repositório seja acessado, tanto local quanto em uma máquina da mesma rede; 4) Manter fonte de dados. Esse requisito permite criar, editar e remover fontes de dados; 5) Manter cubo de dados. Esse requisito permite criar, editar e remover cubos de dados; 6) Manter fato. Esse requisito permite a adição e remoção de fatos do cubo de dados; 7) Manter dimensão. Esse requisito permite a adição e remoção de dimensões do cubo de dados; 8) Manter tabela. Esse requisito permite a adição e remoção de tabelas do cubo de dados; 9) Manter atributo de fato. Esse requisito permite a adição, remoção e a alteração da visibilidade de atributos dos fatos do cubo de dados; 10) Manter atributo de dimensão. Esse requisito permite a adição, remoção e a alteração da visibilidade de atributos das dimensões do cubo de dados; 11) Manter relacionamento. Esse requisito permite a criação e remoção de relacionamentos entre o fato e as dimensões

Requisitos Funcionais
R1: Criar Repositório
R2: Criar Link para Repositório
R3: Abrir Repositório
R4: Manter Fonte de Dados (Criar, Editar e Remover)
R5: Manter Cubo de Dados (Criar, Editar e Remover)
R6: Manter Fato (Adicionar e Remover)
R7: Manter Dimensão (Adicionar e Remover)

R8: Manter Tabela (Adicionar e Remover)
R9: Manter Atributo de Fato (Adicionar, Remover e Alterar Visibilidade)
R10: Manter Atributo de Dimensão (Adicionar, Remover e Alterar Visibilidade)
R11: Manter Relacionamento (Adicionar e Remover)
<b>Requisitos Não-Funcionais</b>
R12: Autenticação de usuário
R13: A ferramenta deve ser usada em desktop
R14: O SQL Server precisa estar instalado (TCP/IP Habilitado)
R15: Linguagem de Programação Java

**Quadro 1 Requisitos funcionais e não funcionais da ferramenta**

#### **4.1.2. Casos de Uso**

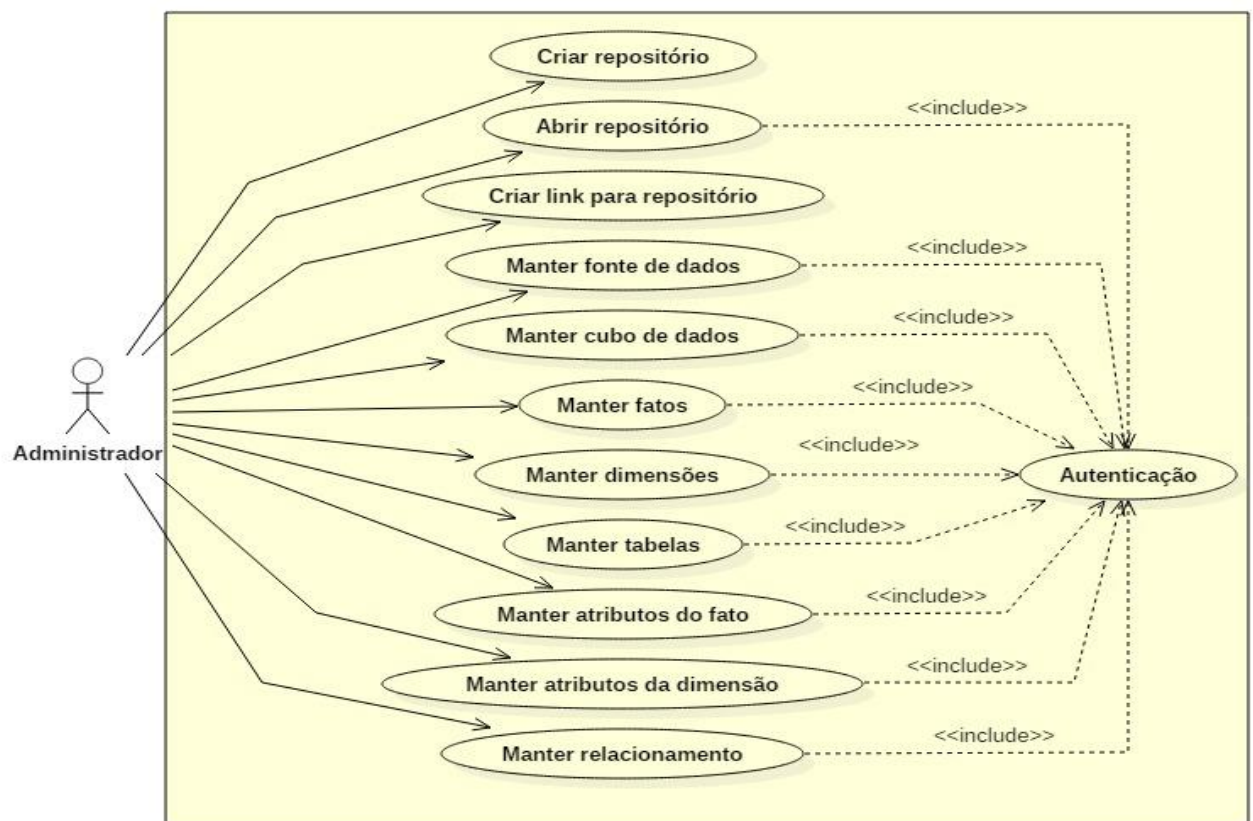
Esta seção descreve os casos de uso em nível de usuário exibindo diagramas que ajudam a entender os atores destinados para cada funcionalidade do sistema. Além disso, ainda nesta seção é descrito os casos de uso em nível de sistema, que auxiliam na compreensão de cada funcionalidade.

##### **4.2.2.1.Casos de Uso – Nível de Usuário**

Após a identificação e análise inicial dos requisitos essenciais da ferramenta, foi modelado o diagrama de casos de uso, que é mostrado na Figura 6. Com ele, é possível



verificar as suas funções.



**Figura 7 Diagrama de casos de uso da ferramenta de administrador**

O caso de uso Criar Repositório tem como objetivo realizar a criação de um repositório de dados. A ação pode acontecer de modo local e também de forma remota. Este caso de uso está associado ao requisito R1 da seção 4.1.1.

O caso de uso Abrir Repositório tem como objetivo realizar o acesso a um repositório de dados. A ação pode acontecer de modo local e também em uma máquina que esteja na mesma rede que a máquina que está sendo executada a ferramenta. Este caso de uso está associado ao requisito R2 da seção 4.1.1.

O caso de uso Criar Link para Repositório tem como objetivo realizar a criação de um link que permita o acesso a um repositório de dados. A ação pode acontecer em um repositório de dados que esteja local e também para acessar uma máquina que esteja na mesma rede que a máquina que está sendo executada a ferramenta. Este caso de uso está associado ao requisito R3 da seção 4.1.1.

O caso de uso Manter Fonte de Dados tem como objetivo realizar a manutenção de fontes de dados, podendo criar, editar e remover. Este caso de uso está associado ao requisito R4 da seção 4.1.1.

O caso de uso Manter Cubo de Dados tem como objetivo realizar a manutenção de cubo de dados, podendo criar, editar e remover. Este caso de uso está associado ao requisito R5 da seção 4.1.1.

O caso de uso Manter Fato tem como objetivo realizar a manutenção de fatos, podendo adicionar e remover. Este caso de uso está associado ao requisito R6 da seção 4.1.1.

O caso de uso Manter Dimensão tem como objetivo realizar a manutenção de dimensões, podendo adicionar e remover. Este caso de uso está associado ao requisito R7 da seção 4.1.1.

O caso de uso Manter Tabela tem como objetivo realizar a manutenção das tabelas, podendo adicionar e remover. Este caso de uso está associado ao requisito R8 da seção 4.1.1.

O caso de uso Manter Atributo do Fato tem como objetivo realizar a manutenção dos atributos dos fatos, podendo adicionar, remover e alterar a sua visibilidade para o seu uso. Este caso de uso está associado ao requisito da seção 4.1.1.

O caso de uso Manter Atributo da Dimensão tem como objetivo realizar a manutenção dos atributos as dimensões, podendo adicionar, remover e alterar a sua visibilidade para o seu uso. Este caso de uso está associado ao requisito da seção 4.1.1.

O caso de uso Manter Relacionamento tem como objetivo realizar a manutenção dos relacionamentos, podendo adicionar e remover. Este caso de uso está associado ao requisito R11 da seção 4.1.1.

#### **4.2.2.2.Casos de Uso – Nível de Sistema**

Nesta seção apresentaremos os casos de uso em nível de sistema. Nestes são apresentados o objetivo, os atores, pré-condição e pós-condição para realização do caso de uso. Além disso, os fluxos normal, alternativos e de exceção são detalhados para conhecimento das interações de cada funcionalidade do sistema.

#### 4.2.2.2.1. Caso de Uso Criar Repositório

<b>Objetivo:</b>	Criar um repositório de dados na máquina em que está sendo executada a ferramenta.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Nenhuma.
<b>Fluxo normal:</b>	1: Clica na opção de criar repositório. 2: Preenche o formulário com as informações solicitadas. 3: Clica no botão criar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não cria se existirem campos vazios. - Não cria se já existir um repositório com o mesmo nome na máquina. - Não se o login ou senha para o SQLServer estiverem errados.

**Quadro 2 Caso de Uso Criar Repositório**

#### 4.2.2.2.2. Caso de Uso Criar Link Para Repositório

<b>Objetivo:</b>	Criar o link para um repositório de dados na máquina local ou em outra máquina da rede.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Nenhuma.
<b>Fluxo normal:</b>	1: Clica na opção de criar link para repositório. 2: Preenche o formulário com as informações solicitadas. 3: Clica no botão criar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não cria se existirem campos vazios.

**Quadro 3 Caso de Uso Criar Link Para Repositório**

#### 4.2.2.2.3. Caso de Uso Abrir Repositório

<b>Objetivo:</b>	Abrir um repositório de dados na máquina local ou em outra máquina da rede para criação ou edição de um cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	- Um repositório precisa existir. - Um link para o repositório precisa existir.
<b>Fluxo normal:</b>	1: Clica na opção de abrir repositório. 2: Informa o login e a senha. 3: Clica no botão entrar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de Exceção:</b>	- Não abre se login e/ou senha estiverem errados. - Não abre se não existir repositório criado. - Não abre se não existir um link para repositório criado.

**Quadro 4 Abrir Repositório**

#### 4.2.2.2.4. Caso de Uso Manter fonte de dados (criar, editar e remover)

**Seção:** criar fonte de dados

<b>Objetivo:</b>	Criar uma fonte de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Nenhuma.
<b>Fluxo normal:</b>	1: Clica na opção de criar fonte de dados. 2: Preenche o formulário com as informações solicitadas. 3: Clica no botão salvar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não cria se existirem campos vazios no formulário. - Não cria se já existir uma fonte de dados com o mesmo nome no repositório.

**Quadro 5 Criar fonte de dados**

**Seção:** editar fonte de dados

<b>Objetivo:</b>	Editar informações da fonte de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Fontes de dados precisam existir.
<b>Fluxo normal:</b>	1: Clica na opção de editar fonte de dados. 2: Clicando, seleciona a fonte de dados que deseja editar. 3: Clica no botão de editar. 4: Confirma a operação. 5: Preenche o formulário com as informações solicitadas. 6: Clica no botão salvar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não conclui a edição se existirem campos vazios no formulário. - Não edita se já existir uma fonte de dados com o novo nome no repositório.

**Quadro 6 Editar fonte de dados**

**Seção:** remover fonte de dados

<b>Objetivo:</b>	Remover fonte de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	- Fontes de dados precisam existir. - Fonte de dados não pode estar sendo usada.
<b>Fluxo normal:</b>	1: Clica na ação de editar fonte de dados. 2: Seleciona a fonte de dados que deseja remover. 3: Clica no botão de remover. 4: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de Exceção:</b>	- Não remove se a fonte de dados tiver sendo usada

**Quadro 7 Remover fonte de dados**

#### 4.2.2.2.5. Caso de Uso Manter cubo de dados (criar, editar e remover)

**Seção:** criar cubo de dados

<b>Objetivo:</b>	Criar um cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir uma fonte de dados.
<b>Fluxo normal:</b>	1: Clica na opção de criar cubo de dados. 2: Seleciona uma fonte de dados. 3: Preenche o formulário com as informações solicitadas. 4: Clica no botão iniciar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não cria se o campo de nome do cubo estiver vazio. - Não cria se já existir um cubo de dados com o mesmo nome no repositório. - Não cria se não existir fontes de dados.

**Quadro 8 Criar cubo de dados**

**Seção:** editar cubo de dados

<b>Objetivo:</b>	Editar informações do cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Cubos de dados precisam existir.
<b>Fluxo normal:</b>	1: Clica na opção de editar cubo de dados. 2: Seleciona o cubo de dados que deseja editar. 3: Clica no botão de editar. 4: Confirma a operação. 5: Preenche o formulário com as informações solicitadas. 6: Clica no botão atualiza.
<b>Fluxo alternativo:</b>	- Não atualiza em caso de falhas. - Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não conclui a edição se o campo de nome do cubo estiver vazio. - Não edita se já existir um cubo de dados com o novo nome no repositório. - Não edita se não existir fontes de dados.

**Quadro 9 Editar cubo de dados**

**Seção:** remover cubo de dados

<b>Objetivo:</b>	Remover cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	- Cubos de dados precisam existir. - Fonte de dados do cubo precisa não mais existir.
<b>Fluxo normal:</b>	1: Clica na opção de editar cubo de dados. 2: Seleciona o cubo de dados que deseja remover. 3: Clica no botão de remover. 4: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não remove se a fonte de dados do cubo de dados existir.

#### Quadro 10 Remover cubo de dados

##### 4.2.2.2.6. Caso de Uso Manter fato (adicionar e remover)

Seção: adicionar fato

<b>Objetivo:</b>	Identificar uma tabela como fato e adicioná-lo no cubo.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir tabela no cubo.
<b>Fluxo normal:</b>	1: Clica em uma tabela exibida. 2: Dá um nome para ela ser chamada como um fato. 3: Clica no botão adicionar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação. - Permite avançar para os próximos passos.
<b>Fluxo de exceção:</b>	- Não adiciona se já existir um fato no cubo de dados com o mesmo nome. - Não adiciona se não for dado um nome para o fato.

#### Quadro 11 Adicionar fato

Seção: remover fato

<b>Objetivo:</b>	Remover um fato do cubo.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir fato no cubo.
<b>Fluxo normal:</b>	1: Clica em um fato exibido. 2: Clica no botão remover. 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite avançar para os próximos passos. - Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

#### Quadro 12 Remover fato

##### 4.2.2.2.7. Caso de Uso Manter dimensão (adicionar e remover)

Seção: adicionar dimensão

<b>Objetivo:</b>	Identificar uma tabela como dimensão e adicioná-la no cubo.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir tabela no cubo.
<b>Fluxo normal:</b>	1: Clica em uma tabela exibida. 2: Dá um nome para ela ser chamada como uma dimensão. 3: Clica no botão adicionar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação. - Permite avançar para os próximos passos.
<b>Fluxo de exceção:</b>	- Não adiciona se já existir uma dimensão no cubo de dados com o mesmo nome. - Não adiciona se não for dado um nome para a dimensão.

#### Quadro 13 Adicionar dimensão

**Seção:** remover dimensão

<b>Objetivo:</b>	Remover uma dimensão do cubo.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir dimensão no cubo.
<b>Fluxo normal:</b>	1: Clica em uma dimensão exibida. 2: Clica no botão remover. 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite avançar para os próximos passos. - Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 14 Remover dimensão**

**4.2.2.2.8. Caso de Uso Manter Tabela (adicionar e remover)**

**Seção:** adicionar tabela

<b>Objetivo:</b>	Adicionar novas tabelas ao cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir novas tabela.
<b>Fluxo normal:</b>	1: Clica no botão atualizar.
<b>Fluxo alternativo:</b>	- Permite avançar para os próximos passos.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 15 Adicionar tabela**

**Seção:** remover tabela

<b>Objetivo:</b>	Remover uma tabela do cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir tabela no cubo.
<b>Fluxo normal:</b>	1: Clica em uma tabela exibida. 2: Clica no botão remover.
<b>Fluxo alternativo:</b>	- Permite avançar para os próximos passos.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 16 Remover tabela**

**4.2.2.2.9. Caso de Uso Manter atributo do fato (adicionar, remover e alterar visibilidade)**

**Seção:** adicionar atributo do fato

<b>Objetivo:</b>	Adicionar um atributo do fato ao cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo na tabela identificada como fato.
<b>Fluxo normal:</b>	1: Clica em um fato exibido. 2: Clica em um atributo. 3: Dá um nome para o atributo.

	4: Escolhe função de agregação. 5: Clica no botão adicionar.
<b>Fluxo alternativo:</b>	- Permite ir a próximos passos ou passos anteriores.
<b>Fluxo de exceção:</b>	- Não adiciona se já existir um atributo com o mesmo nome.

**Quadro 17 Adicionar atributo do fato**

**Seção:** remover atributo do fato

<b>Objetivo:</b>	Remover um atributo do fato.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo no fato.
<b>Fluxo normal:</b>	1: Clica em um atributo do fato exibido. 2: Clica no botão remover. 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite ir a próximos passos ou passos anteriores. - Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 18 Remover atributo do fato**

**Seção:** alterar visibilidade

<b>Objetivo:</b>	Alterar a visibilidade do atributo do fato na tela do usuário.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo no fato.
<b>Fluxo normal:</b>	1: Clica em um atributo do fato exibido. 2: Clica no botão alterar.
<b>Fluxo alternativo:</b>	- Permite ir a próximos passos ou passos anteriores.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 19 Alterar visibilidade**

#### 4.2.2.2.10. Caso de Uso Manter atributo da dimensão (adicionar, remover e alterar visibilidade)

**Seção:** adicionar atributo da dimensão

<b>Objetivo:</b>	Tem como objetivo adicionar um atributo da dimensão ao cubo de dados.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo na tabela identificada como dimensão.
<b>Fluxo normal:</b>	1: Clica em uma dimensão exibida. 2: Clica em um atributo. 3: Dá um nome para o atributo. 4: Clica no botão adicionar.
<b>Fluxo alternativo:</b>	- Permite ir a próximos passos ou passos anteriores.



<b>Fluxo de exceção:</b>	- Não adiciona se já existir um atributo com o mesmo nome.
--------------------------	--

**Quadro 20 Adicionar atributo da dimensão**

**Seção:** remover atributo da dimensão

<b>Objetivo:</b>	Tem como objetivo remover um atributo da dimensão.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo na dimensão.
<b>Fluxo normal:</b>	1: Clica em um atributo da dimensão exibida. 2: Clica no botão remover. 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite ir a próximos passos ou passos anteriores. - Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 21 Remover atributo da dimensão**

**Seção:** alterar visibilidade

<b>Objetivo:</b>	Alterar a visibilidade do atributo da dimensão na tela do usuário.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo na dimensão.
<b>Fluxo normal:</b>	1: Clica em um atributo da dimensão exibido. 2: Clica no botão alterar.
<b>Fluxo alternativo:</b>	- Permite ir a próximos passos ou passos anteriores.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 22 Alterar visibilidade**

#### 4.2.2.2.11. Caso de Uso Manter relacionamento (adicionar e remover)

**Seção:** adicionar relacionamento

<b>Objetivo:</b>	Adicionar um relacionamento entre um fato e uma dimensão.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir atributo no fato e na dimensão.
<b>Fluxo normal:</b>	1: Seleciona um fato. 2: Seleciona um atributo do fato. 3: Seleciona uma dimensão. 4: Seleciona um atributo da dimensão. 5: Dá um nome para o relacionamento. 6: Clica no botão adicionar.
<b>Fluxo alternativo:</b>	- Permite ir aos passos anteriores ou finalizar.
<b>Fluxo de exceção:</b>	- Não adiciona se já existir um relacionamento com o mesmo nome. - Não adiciona se já existe o relacionamento.

**Quadro 23 Adicionar relacionamento**

### Seção: remover relacionamento

<b>Objetivo:</b>	Remover um relacionamento entre um fato e uma dimensão.
<b>Atores:</b>	Administrador.
<b>Pré-condição:</b>	Precisa existir relacionamento criado.
<b>Fluxo normal:</b>	1: Seleciona um relacionamento exibido. 2: Clica no botão remover 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite ir aos passos anteriores ou finalizar. - Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

Quadro 24 Remover relacionamento

### 4.1.3. Diagramas de Classe

Após a criação dos casos de uso da ferramenta, foram modelados os diagramas de classes do Sistema. Nesta seção apresentaremos os diagramas separados por casos de uso, detalhando as classes que o sistema necessita possuir e os relacionamentos entre si.

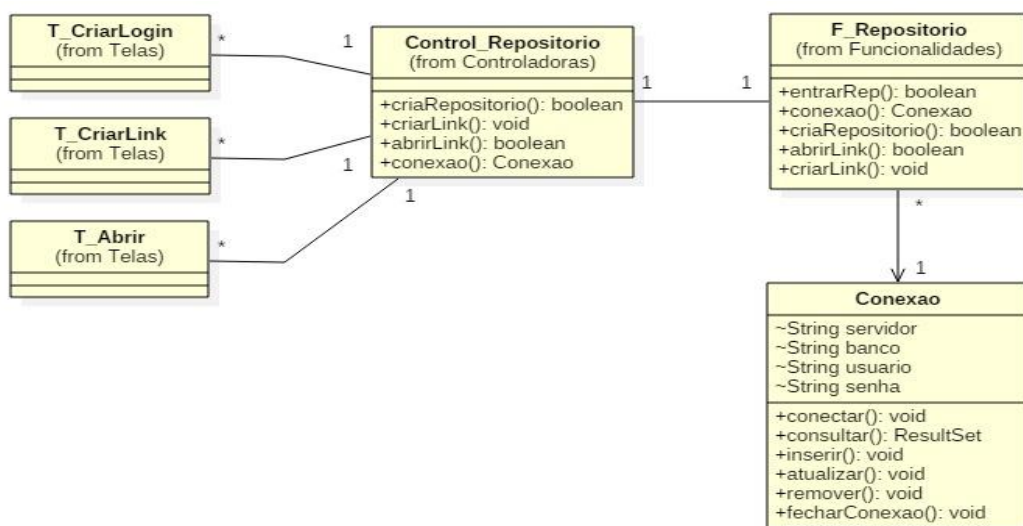


Figura 8 Diagrama de classes dos casos de uso das seções 4.1.3.1, 4.1.3.2 e 4.2.3.3

#### 4.2.3.1.Caso de Uso Criar Repositório

**T\_CriarLogin** – Representa a interface na qual o administrador passar os dados necessários para a criação do repositório e do seu login.

**Control\_Repositorio** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Repositorio** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos `criaRepositorio()` e `criaLink()`, para criar o repositório

que será usado para inserir os dados da fonte de dados e para criar o link necessário para acessar o mesmo.

**Conexão:** Representa a classe responsável para a conexão com o banco de dados.

#### **4.2.3.2.Caso de Uso Criar Link para Repositório**

**T\_CriarLink** – Representa a interface na qual o administrador passar os dados necessários para a criação do link para acessar o repositório.

**Control\_Repositorio** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Repositorio** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utilizando o método *criaLink()*, para criar o link necessário para acessar o repositório.

**Conexão:** Representa a classe responsável para a conexão com o banco de dados.

#### **4.2.3.3.Caso de Uso Abrir Repositório**

**T\_Abrir** – Representa a interface na qual o administrador passar os dados necessários para a abrir o repositório.

**Control\_Repositorio** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Repositorio** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *abrirLink()* e *entraRep()*, para abrir o repositório de acordo com o link e para verificar o login a senha do repositório, respectivamente.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### **4.2.3.4.Caso Manter Fonte de Dados (criar, editar e remover)**

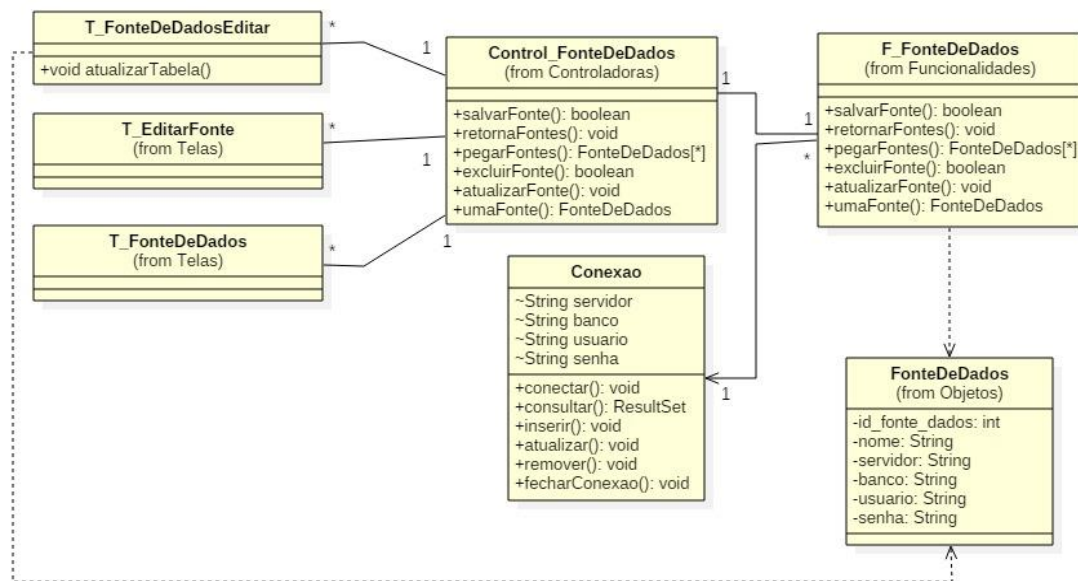


Figura 9 Diagrama de classes do caso de uso manter fonte de dados

**T\_FonteDeDadosEditar** – Representa a interface na qual o administrador passar os dados necessários para a atualizar a fonte de dados.

**T\_EditarFonte** - Representa a interface na qual o administrador seleciona a fonte de dados que se quer editar ou remover.

**T\_FonteDeDados** - Representa a interface onde o administrador passar os dados necessários para a criar uma fonte de dados.

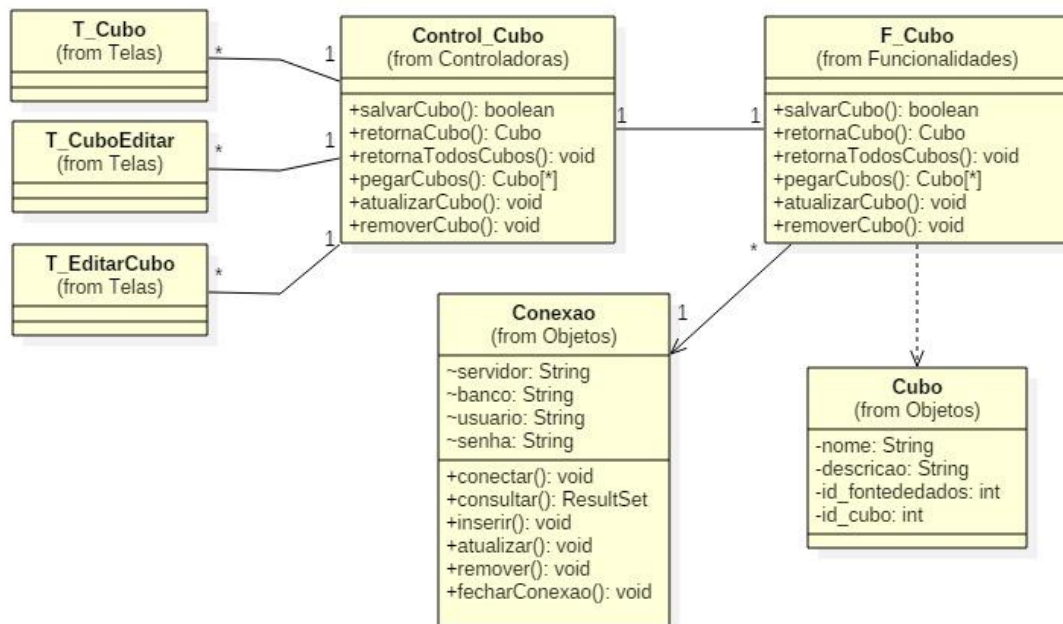
**Control\_FonteDeDados** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_FonteDeDados** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *salvarFonte()* para salvar a fonte de dados que foi criada, *retornaFontes()* e *pegarFontes()* para carregar os vetores com as fonte que estão sendo chamadas no momento da execução, *excluirFonte()* para remover uma fonte de dados, *atualizaFonte()* para atualizar as informações de uma fonte de dados e *umaFonte()* para carregar as informações da fonte selecionada no momento da criação do cubo.

**FonteDeDados** - Classe que corresponde aos dados da fonte de dados.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.5.Caso Manter Cubo de Dados (criar, editar e remover)



**Figura 10 Diagrama de classes do caso de uso manter cubo de dados**

**T\_CuboEditar** – Representa a interface na qual o administrador passar os dados necessários para a atualizar o cubo de dados.

**T\_EditarCubo** - Representa a interface na qual o administrador seleciona o cubo de dados que se quer editar ou remover.

**T\_Cubo** - Representa a interface onde o administrador passar os dados necessários para a criar um cubo de dados.

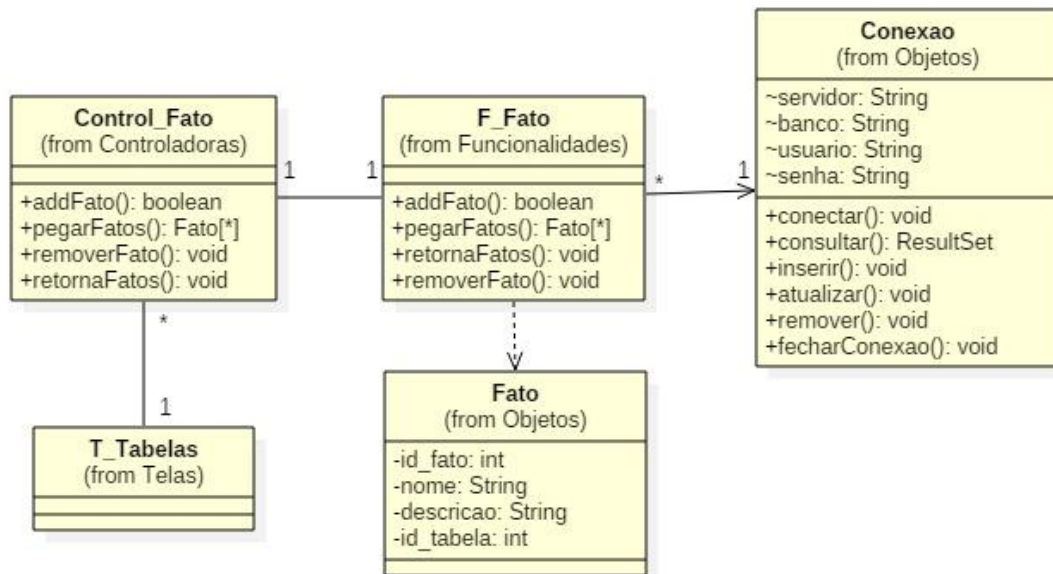
**Control\_Cubo** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Cubo** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *salvarCubo()* para salvar o cubo de dados que foi criado, *retornarCubos()* e *pegarCubos()* para carregar os vetores com os cubos que estão sendo chamadas no momento da execução, *removerCubo()* para remover um cubo de dados, *atualizaCubo()* para atualizar as informações de um cubo de dados e *retornarTodosCubos()* para carregar todos os cubos cadastrados no repositório.

**Cubo** - Classe que corresponde aos dados do cubo de dados.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.6.Caso de Uso Manter fato (adicionar e remover)



**Figura 11 Diagrama de classes do caso de uso manter fato**

**T\_Tabelas** – Representa a interface na qual o administrador visualiza as tabelas que são identificadas como fato.

**Control\_Fato** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Fato** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *addFato()* para salvar o fato no cubo de dados, *retornaFatos()* e *pegarFatos()* para carregar os vetores com os fatos que estão sendo chamados no momento da execução e *removerFato()* para remover o fato do cubo de dados.

**Fato** - Classe que corresponde aos dados do fato.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### **4.2.3.7.Caso de Uso Manter dimensão (adicionar e remover)**

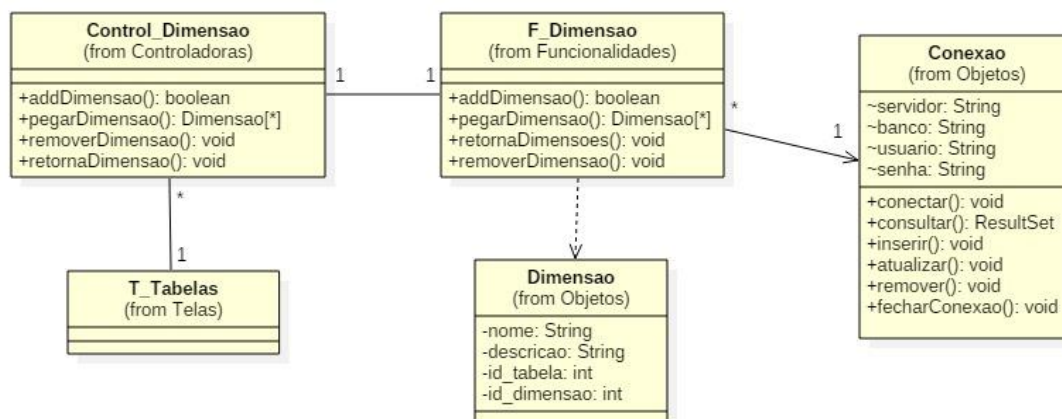


Figura 12 Diagrama de classes do caso de uso manter dimensão

**T\_Tabelas** – Representa a interface na qual o administrador visualiza as tabelas que são identificadas como dimensão.

**Control\_Dimensão** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Dimensão** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *addDimensão()* para salvar a dimensão no cubo de dados, *retornaDimensoes()* e *pegarDimensao()* para carregar os vetores com as dimensões que estão sendo chamadas no momento da execução e *removerDimensao()* para remover a dimensão do cubo de dados.

**Dimensão** - Classe que corresponde aos dados da dimensão.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.8.Caso de Uso Manter tabela (adicionar e remover)

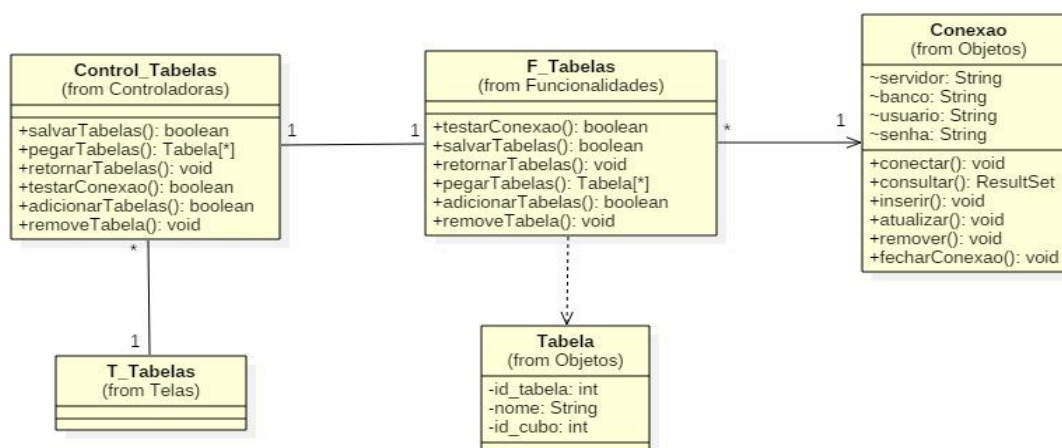


Figura 13 Diagrama de classes do caso de uso manter tabela



**T\_Tabelas** – Representa a interface na qual o administrador visualiza as tabelas da fonte de dados.

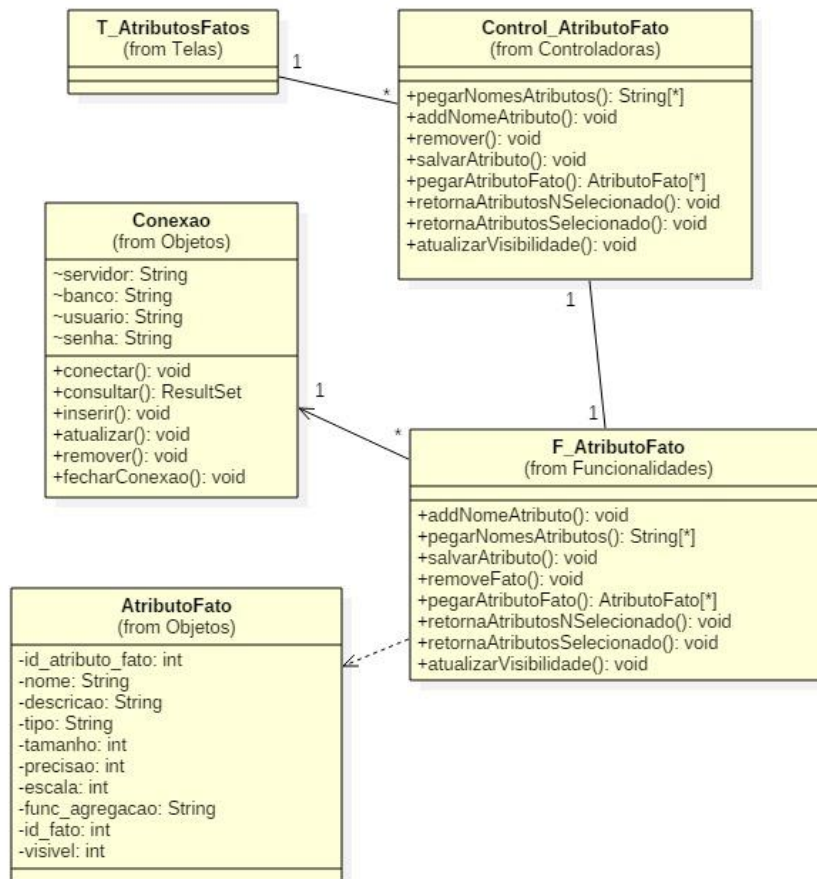
**Control\_Tabelas** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Tabelas** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *salvarTabelas()* para salvar as tabelas no cubo de dados, *retornaTabelas()* e *pegarTabelas()* para carregar os vetores com as tabelas que estão sendo chamadas no momento da execução, *removeTabelaDimensao()* para remover uma tabela, *adicionarTabelas()* para adicionar novas tabelas no cubo de dados e *testarConexao()* para testar a conexão da fonte que será utilizada para visualização das tabelas.

**Tabela** - Classe que corresponde aos dados da dimensão.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.9.Caso de Uso Manter Atributo do Fato



**Figura 14** Diagrama de classes do caso de uso manter atributo do fato



**T\_AtributosFatos** – Representa a interface na qual o administrador escolhe os atributos que se tornarão atributos do fato.

**Control\_AtributoFato** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_AtributoFato** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *salvarAtributo()* para salvar os atributos do fato no cubo de dados, *addNomeAtributo()*, *pegarNomesAtributos()* e *pegarAtributoFato()* para carregar os vetores com os atributos do fato e os seus nomes que estão sendo chamados no momento da execução, *remover()* para remover o atributo do fato, *atualizarVisibilidade()* para alterar a visibilidade do atributo na ferramenta do usuário, *retornaAtributosNSelecioneado()* para carregar os vetores com os atributos escolhidos e *retornaAtributosNSelecioneado()* para carregar os vetores com os atributos não escolhidos.

**AtributoFato** - Classe que corresponde aos dados do atributo do fato.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.10. Caso de Uso Manter Atributo da Dimensão

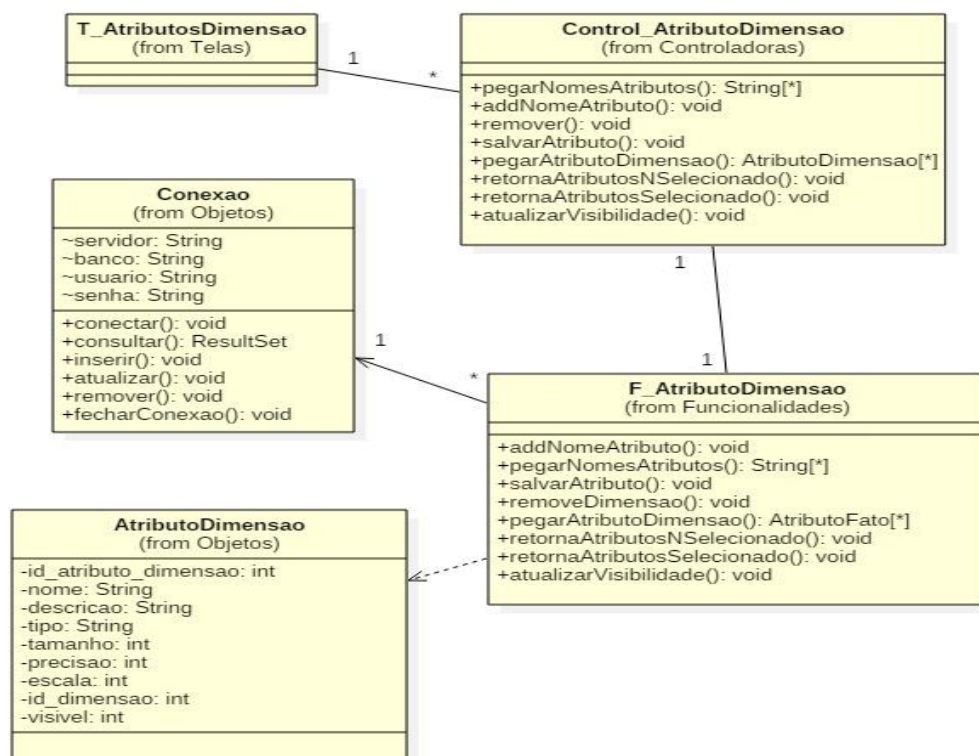


Figura 15 Diagrama de classes do caso de uso manter atributo da dimensão

**T\_AtributosDimensao** – Representa a interface na qual o administrador escolhe os atributos que se tornarão atributos da dimensão.

**Control\_AtributoDimensao** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_AtributoDimensao** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *salvarAtributo()* para salvar os atributos da dimensão no cubo de dados, *addNomeAtributo()*, *pegarNomesAtributos()* e *pegarAtributoDimensao()* para carregar os vetores com os atributos da dimensão e os seus nomes que estão sendo chamados no momento da execução, *remover()* para remover o atributo da dimensão, *atualizarVisibilidade()* para alterar a visibilidade do atributo na ferramenta do usuário, *retornaAtributosNSelecioneado()* para carregar os vetores com os atributos escolhidos e *retornaAtributosNSelecioneado()* para carregar os vetores com os atributos não escolhidos.

**AtributoDimensao** - Classe que corresponde aos dados do atributo da dimensão.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.11. Caso de Uso Manter Relacionamento

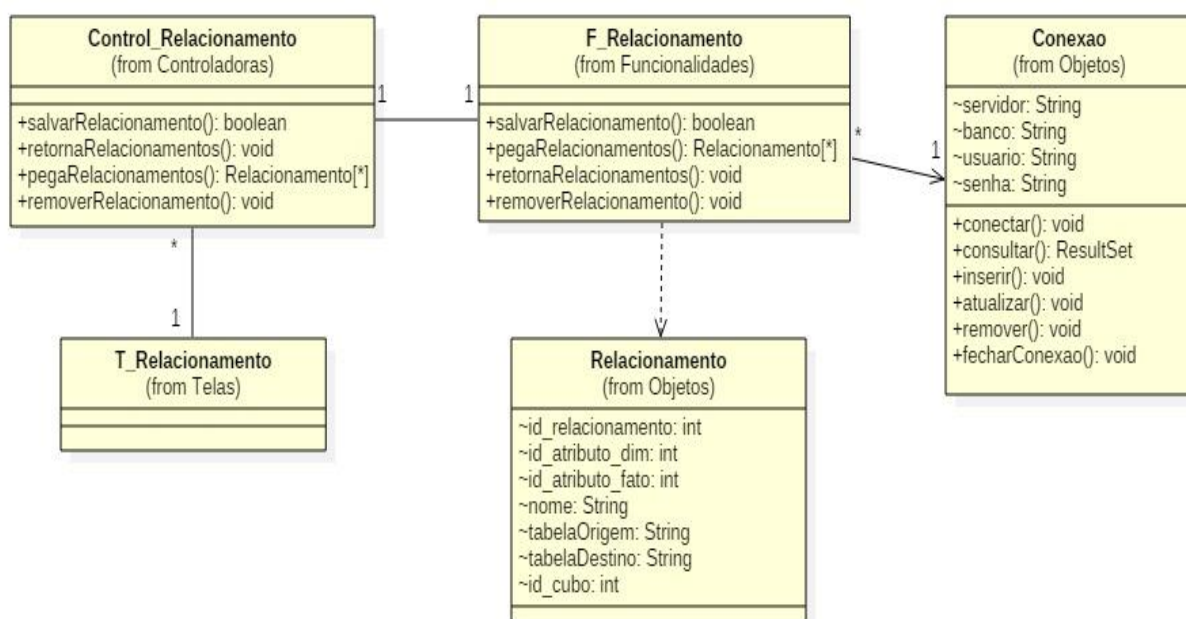


Figura 16 Diagrama de classes do caso de uso manter relacionamento

**T\_Relacionamento** – Representa a interface na qual o administrador escolhe os atributos do fato e da dimensão que irão se relacionar.

**Control\_Relacionamento** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Relacionamento** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *salvarRelacionamento()* para salvar os relacionamento que são criados entre o fato e a dimensão, *pegaRelacionamento()* e *retornaRelacionamentoDimensao()* para carregar os vetores com os relacionamentos que estão sendo chamados no momento da execução e *removerRelacionamento()* para remover um relacionamento.

**Relacionamento** - Classe que corresponde aos dados do relacionamento.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

## 4.2. Ferramenta de Usuário (Módulo de criação de consultas)

### 4.2.1. Requisitos funcionais e não funcionais

Os requisitos funcionais da ferramenta, resumidos no Quadro 2, são: 1) Abrir repositório. Esse requisito permite que um repositório seja acessado; 2) Selecionar Cubo. Esse requisito permite que um cubo seja escolhido para trabalhar; 3) Criar consulta. Esse requisito permite que consultas sejam criadas, adicionando e removendo atributos e adicionando e removendo cláusulas.

Requisitos Funcionais
R1: Abrir Repositório
R2: Selecionar Cubo
R3: Criar consulta (Adicionar Atributo, Remover Atributo, Adicionar Cláusula e Remover Cláusula)
Requisitos Não-Funcionais
R12: Autenticação de usuário
R13: A ferramenta deve ser usada em desktop
R14: O SQL Server precisa estar instalado (TCP/IP Habilitado)
R15: Linguagem de Programação Java

**Quadro 25** Requisitos funcionais e não funcionais da ferramenta

#### 4.2.2. Casos de Uso

Esta seção descreve os casos de uso em nível de usuário exibindo diagramas que ajudam a entender os atores destinados para cada funcionalidade do sistema. Além disso, ainda nesta seção são descritos os casos de uso em nível de sistema, que auxiliam na compreensão de cada funcionalidade.

##### 4.2.2.1. Casos de Uso – Nível de Usuário

Após a identificação e análise inicial dos requisitos essenciais da ferramenta, foi modelado o diagrama de casos de uso, que é mostrado na Figura 17. Com ele, é possível verificar as suas funções.



**Figura 17 Diagrama de casos de uso da ferramenta de usuário**

O caso de uso Abrir Repositório tem como objetivo realizar o acesso a um repositório de dados. A ação pode acontecer de modo local e também de forma remota. Este caso de uso está associado ao requisito R1 da seção 4.2.1.

O caso de uso Selecionar Cubo tem como objetivo selecionar um cubo de dados que será trabalhado na geração das consultas. Este caso de uso está associado ao requisito R2 da seção 4.2.1.

O caso de uso Criar Consulta tem como objetivo criar as consultas que serão executadas, podendo adicionar e remover atributos e adicionar e remover cláusulas. Este caso de uso está associado ao requisito R3 da seção 4.2.1.

##### 4.2.2.2. Casos de Uso – Nível de Sistema

Nesta seção apresentaremos os casos de uso em nível de sistema. Nestes são apresentados o objetivo, os atores, pré-condição e pós-condição para realização do caso de uso. Além disso, os fluxos normal, alternativos e de exceção são detalhados para conhecimento das interações de cada funcionalidade do sistema.

#### 4.2.2.2.1. Caso de Uso Abrir Repositório

<b>Objetivo:</b>	Abrir um repositório de dados na máquina local ou em outra máquina da rede para trabalhar.
<b>Atores:</b>	Usuário.
<b>Pré-condição:</b>	- Um repositório precisa existir. - Um link para o repositório precisa existir.
<b>Fluxo normal:</b>	1: Clica na opção de abrir repositório. 2: Preenche o login e a senha. 3: Clica no botão entrar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	- Não abre se login ou senha estiverem errados. - Não abre se não existir repositório criado. - Não abre se não existir um link para repositório criado.

**Quadro 26 Abrir repositório**

#### 4.2.2.2.2. Caso de Uso Selecionar cubo de dados

<b>Objetivo:</b>	Selecionar um cubo de dados para trabalhar.
<b>Atores:</b>	Usuário.
<b>Pré-condição:</b>	Precisa existir cubo de dados.
<b>Fluxo normal:</b>	1: Clica na opção de selecionar um cubo de dados. 2: Seleciona um cubo de dados. 3: Clica no botão iniciar.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

**Quadro 27 Selecionar cubo de dados**

#### 4.2.2.2.3. Caso de Uso Criar consulta (adicionar atributo, remover atributo, adicionar cláusula, remover cláusula)

**Seção:** adicionar atributo

<b>Objetivo:</b>	Adicionar atributo na consulta.
<b>Atores:</b>	Usuário.
<b>Pré-condição:</b>	Precisa existir um cubo de dados povoado.
<b>Fluxo normal:</b>	1: Clica no atributo do fato ou da dimensão.
<b>Fluxo alternativo:</b>	Nenhum.
<b>Fluxo de exceção:</b>	- Não adiciona se o atributo já tiver sido adicionado.

#### Quadro 28 Adicionar atributo

**Seção:** remover atributo

<b>Objetivo:</b>	Remover atributo na consulta.
<b>Atores:</b>	Usuário.
<b>Pré-condição:</b>	Precisa existir atributos na consulta.
<b>Fluxo normal:</b>	1: Clica no atributo do fato ou da dimensão. 2: Clica no botão remover. 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

#### Quadro 29 Remover atributo

**Seção:** adicionar cláusula

<b>Objetivo:</b>	Adicionar cláusula na consulta.
<b>Atores:</b>	Usuário.
<b>Pré-condição:</b>	Precisa existir um cubo de dados povoado.
<b>Fluxo normal:</b>	1: Clica no atributo. 2: Informa a comparação. 3: Seleciona o operador. 4: Clica em ok.
<b>Fluxo alternativo:</b>	Nenhum.
<b>Fluxo de exceção:</b>	- Não adiciona se já existir a cláusula.

#### Quadro 30 Adicionar cláusula

**Seção:** remover cláusula

<b>Objetivo:</b>	Remover cláusula da consulta.
<b>Atores:</b>	Usuário.
<b>Pré-condição:</b>	Precisa existir cláusula na consulta.
<b>Fluxo normal:</b>	1: Clica na cláusula. 2: Clica no botão remover. 3: Confirma a operação.
<b>Fluxo alternativo:</b>	- Permite cancelar a operação.
<b>Fluxo de exceção:</b>	Nenhum.

#### Quadro 31 Remover cláusula

### 4.2.3. Diagramas de Classe

Após a criação dos casos de uso da ferramenta, foram modelados os diagramas de classes do Sistema. Nesta seção apresentaremos os diagramas separados por casos de uso, detalhando as classes que o sistema necessita possuir e os relacionamentos entre si.

#### 4.2.3.1.Caso de Uso Abrir Repositório

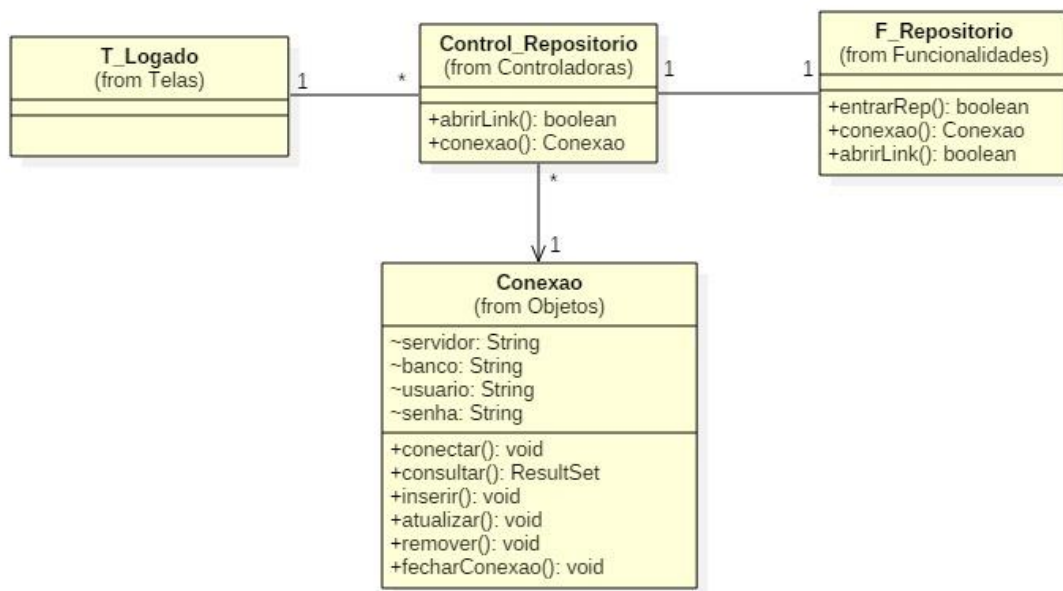


Figura 18 Diagrama de classe do caso de uso abrir repositório

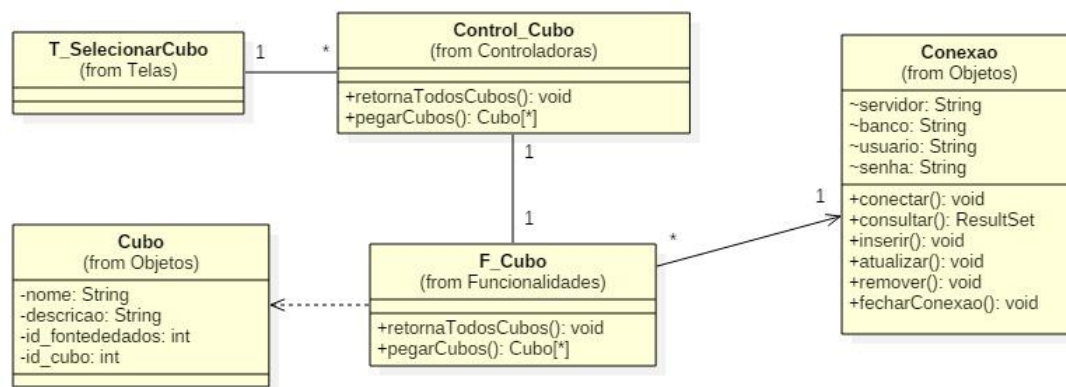
**T\_Logado** – Representa a interface na qual o usuário passar os dados necessários para abrir o repositório.

**Control\_Repositorio** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Repositorio** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *abrirLink()* e *entraRep()*, para abrir o repositório de acordo com o link e para verificar o login a senha do repositório, respectivamente.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.2.3.2.Caso de Uso Selecionar Cubo de Dados



**Figura 19 Diagrama de classe do caso de uso selecionar cubo**

**T\_SelecionarCubo** – Representa a interface na qual o usuário seleciona o cubo de dados que deseja trabalhar.

**Control\_Cubo** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

**F\_Cubo** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *retornaTodosCubos()* e *pegarCubos()*, para carregar os vetores com os relacionamentos que estão sendo chamados no momento da execução.

**Cubo** - Classe que corresponde aos dados do cubo.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### **4.2.3.3.Caso de Uso Criar consulta (adicionar atributos, remover atributos, adicionar cláusula, remover cláusula)**



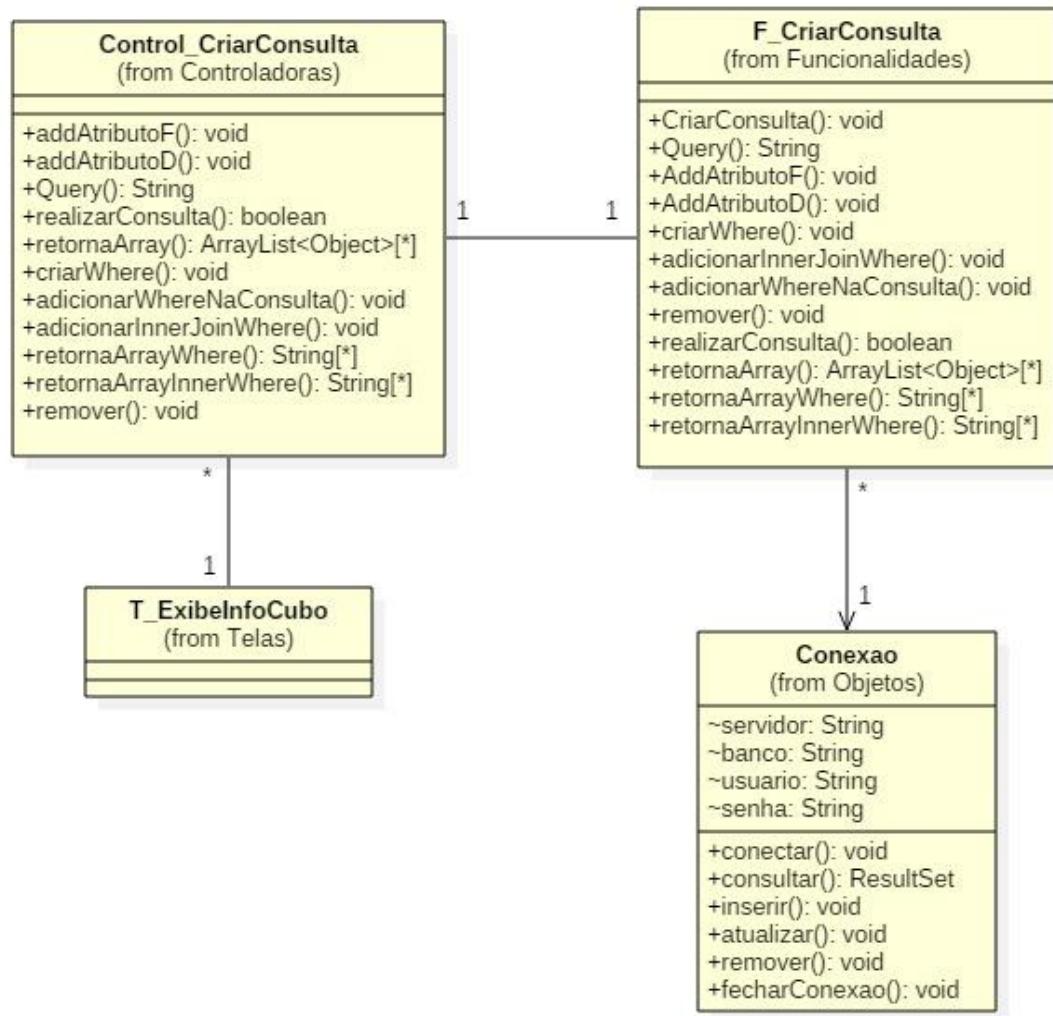


Figura 20 Diagrama de classe do caso de uso criar consulta

**T\_ExibeInfoCubo** – Representa a interface na qual o usuário visualiza todas as informações do cubo de dados que ficam disponíveis para escolha e criar uma consulta.

**Control\_CriaConsulta** – Representa a classe responsável por realizar a troca de dados entre a interface e os métodos funcionais da ferramenta.

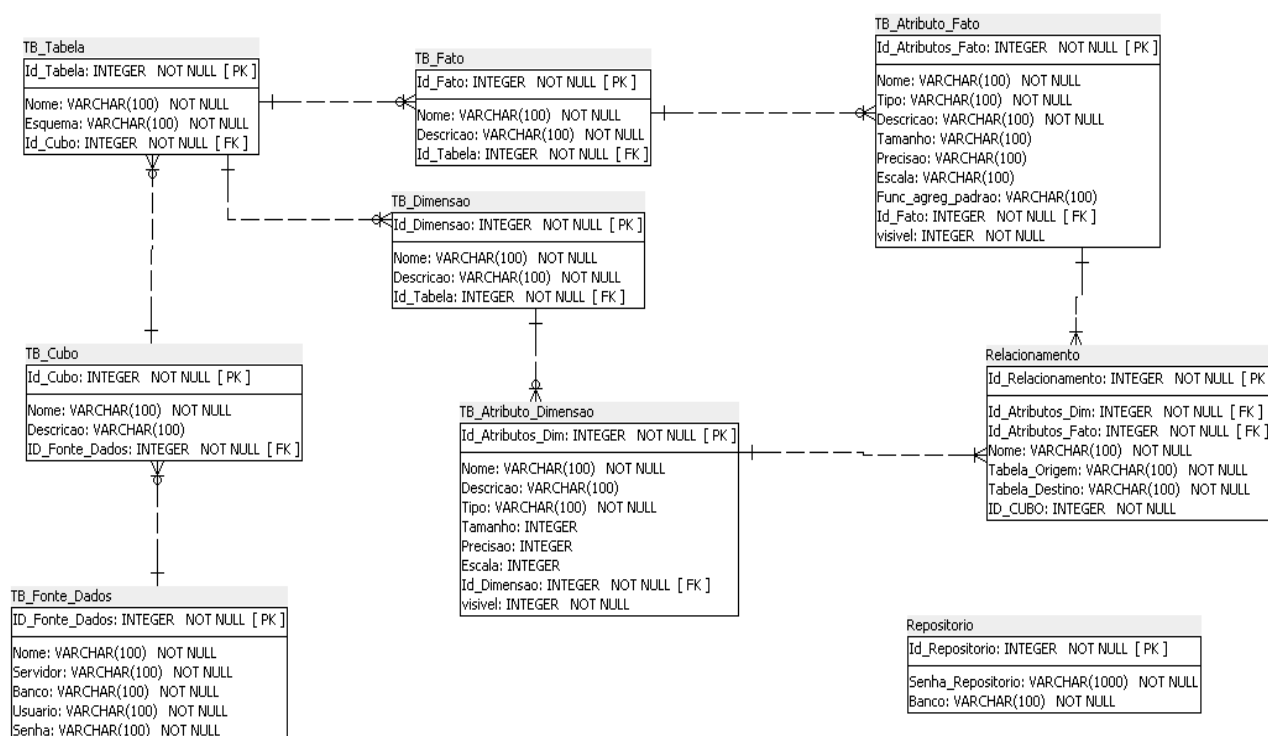
**F\_CriarConsulta** – Representa a classe responsável por todas ações da ferramenta nesse caso de uso. Utiliza os métodos *criarConsulta()* para gerar a consulta que vai ser executada, *Query()* para carregar a consulta que está sendo formada no momento da execução, *AddAtributoF()* e *AddAtributoD()* para adicionar os atributos na consulta, *criarWhere()* para criar a cláusula da consulta e *adicionarInnerJoinWhere()* para colocar a cláusula na consulta, *remover()* para remover algum atributo da consulta, *realizarConsulta()* para enviar a consulta pra ser executada e *retornaArray()*,

*retornaArrayWhere()* e *retornaArrayInnerWhere()*, para carregar os vetores com os atributos e as cláusulas que estão sendo chamados no momento da execução.

**Conexão** - Representa a classe responsável para a conexão com o banco de dados.

#### 4.4. Modelagem de Dados

Após a criação dos diagramas de classe, foi modelado o projeto lógico de dados para o repositório de dados. A Figura 21 mostra o esquema de dados do repositório da ferramenta. Nesta seção detalharemos as tabelas do esquema.



**Figura 21** Esquema de dados da ferramenta

##### 4.4.1. Componentes do modelo de dados

**Tabela tb\_tabela:** Todas as informações sobre as tabelas são persistidas na tabela tb\_tabela. Esta tabela irá armazenar todas as tabelas que existirem na fonte de dados. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Tabela atributo que compõe uma restrição de integridade referencial, um Nome para a tabela e o Esquema. A tabela tb\_tabela ainda possui uma ligação 1-N com a tabela tb\_fato e com a tb\_dimensao, e recebe uma ligação da tabela tb\_cubo. A tb\_tabela tem o atributo Id\_Cubo que é uma chave estrangeira nesta tabela.

**Tabela tb\_cubo:** Todas as informações sobre o cubo de dados são persistidas na tabela tb\_cubo. Esta tabela irá armazenar todos os cubos de dados que forem criados. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Cubo atributo que compõe uma restrição de integridade referencial, um Nome e uma descrição para o cubo, sendo essa última algo optativo. A tabela tb\_cubo ainda possui uma ligação 1-N com a tabela tb\_tabela e recebe uma ligação da tabela tb\_fonte\_dados. A tb\_cubo tem o atributo Id\_Fonte\_Dados que é uma chave estrangeira nesta tabela.

**Tabela tb\_fonte\_dados:** Todas as informações sobre a fonte de dados são persistidas na tabela tb\_fonte\_dados. Esta tabela irá armazenar todas as fontes de dados que forem criadas. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Fonte\_Dados atributo que compõe uma restrição de integridade referencial, um Nome, o Servidor, o Banco, o Usuario e a Senha para o cubo. A tabela tb\_fonte\_dados ainda possui uma ligação 1-N com a tabela tb\_cubo.

**Tabela tb\_fato:** Todas as informações sobre o fato são persistidas na tabela tb\_fato. Esta tabela irá armazenar todos fatos que existirem. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Fato atributo que compõe uma restrição de integridade referencial, um Nome e uma descrição para o fato. A tabela tb\_fato ainda possui uma ligação 1-N com a tabela tb\_atributo\_fato, e recebe uma ligação da tabela tb\_tabela. A tb\_fato tem o atributo Id\_Tabela que é uma chave estrangeira nesta tabela.

**Tabela tb\_dimensao:** Todas as informações sobre a dimensão são persistidas na tabela tb\_dimensao. Esta tabela irá armazenar todas as dimensões que existirem. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Dimensao atributo que compõe uma restrição de integridade referencial, um Nome e uma descrição para a dimensao. A tabela tb\_dimensao ainda possui uma ligação 1-N com a tabela tb\_atributo\_dimensao, e recebe uma ligação da tabela tb\_tabela. A tb\_dimensao tem o atributo Id\_Tabela que é uma chave estrangeira nesta tabela.

**Tabela tb\_atributo\_fato:** Todas as informações sobre o atributo do fato são persistidas na tabela tb\_atributo\_fato. Esta tabela irá armazenar todos os atributos do fato que existirem. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Atributos\_Fato atributo que compõe uma restrição de integridade referencial, um Nome, o Tipo, uma Descrição, o Tamanho, a Precisão, a Escala, a função de agregação em Func\_ereg\_padrao e a visibilidade em Visível, onde 1 significa visível e 0 invisível. A tabela tb\_atributo\_fato ainda possui uma ligação 1-N com a tabela tb\_relacionamento, e recebe uma ligação da tabela tb\_fato. A tb\_atributo\_fato tem o atributo Id\_Fato que é uma chave estrangeira nesta tabela.

**Tabela tb\_atributo\_dimensao:** Todas as informações sobre o atributo da dimensão são persistidas na tabela tb\_atributo\_dimensao. Esta tabela irá armazenar todos os atributos da dimensão que existirem. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_Atributo\_Dim atributo que compõe uma restrição de integridade referencial, um Nome, o Tipo, uma Descrição, o Tamanho, a Precisão, a Escala e a visibilidade em Visível, onde 1 significa visível e 0 invisível. A tabela tb\_atributo\_dimensao ainda possui uma ligação 1-N com a tabela relacionamento, e recebe uma ligação da tabela tb\_dimensao. A tb\_atributo\_dimensao tem o atributo Id\_Dimensao que é uma chave estrangeira nesta tabela.

**Tabela relacionamento:** Todas as informações sobre o relacionamento do fato com a dimensão são persistidas na tabela relacionamento. Esta tabela irá armazenar todos os relacionamentos que existirem. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_relacionamento atributo que compõe uma restrição de integridade referencial, um Nome, o nome do atributo do fato em Tabela\_origem, o nome do atributo da dimensão em Tabela\_destino, e a identificação do cubo que é armazenado em ID\_Cubo. A tabela tb\_relacionamento ainda recebe uma ligação da tabela tb\_atributo\_fato e da tb\_atributo\_dimensao. A tabela relacionamento tem os atributos Id\_atributo\_fato e Id\_atributos\_dim que são chaves estrangeiras nesta tabela.

**Tabela repositorio:** Todas as informações de login sobre o repositório são persistidas na tabela repositorio. Esta tabela irá armazenar todos os logins para os repositórios que

existirem. Logo, cada tabela tem a obrigatoriedade de armazenar um Id\_repositorio atributo que compõe uma restrição de integridade referencial, uma senha em Senha\_Repositorio e um Banco.

## 5. FERRAMENTA

A ferramenta desenvolvida tem como objetivo mostrar as características de uma ferramenta ROLAP de forma didática e intuitiva.

Nesta seção apresentaremos a utilização da ferramenta com um esquema de dados criado para um sistema de *helpdesk*.

O esquema estrela que será utilizado para criação da camada semântica é apresentado na figura 22.

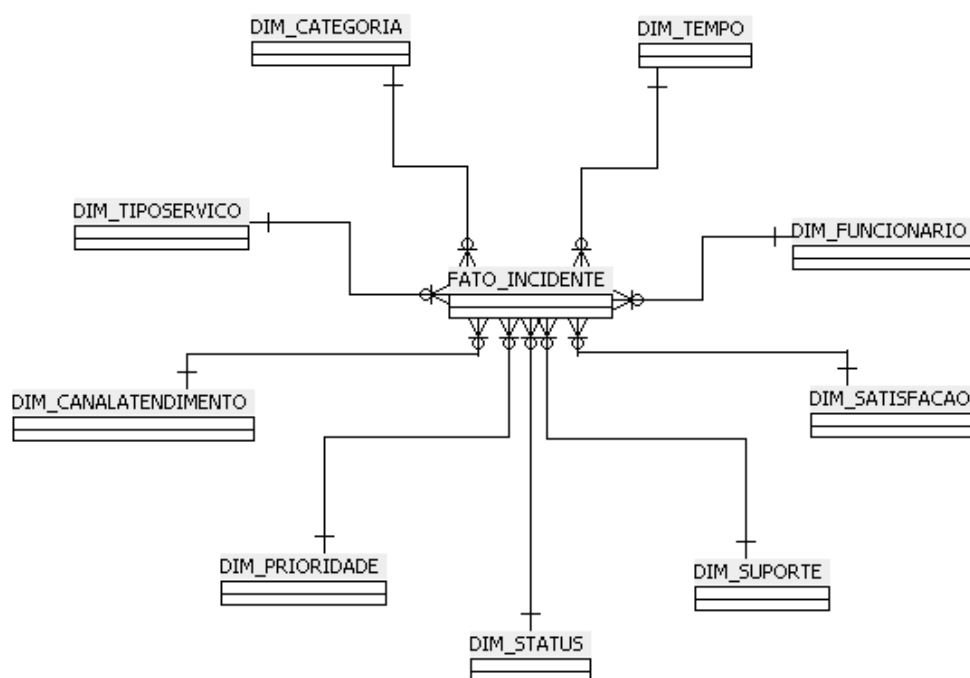


Figura 22 Esquema estrela utilizado

### 5.1.Ferramenta de Administrador (Módulo de configuração do repositório)

#### Tela Inicial

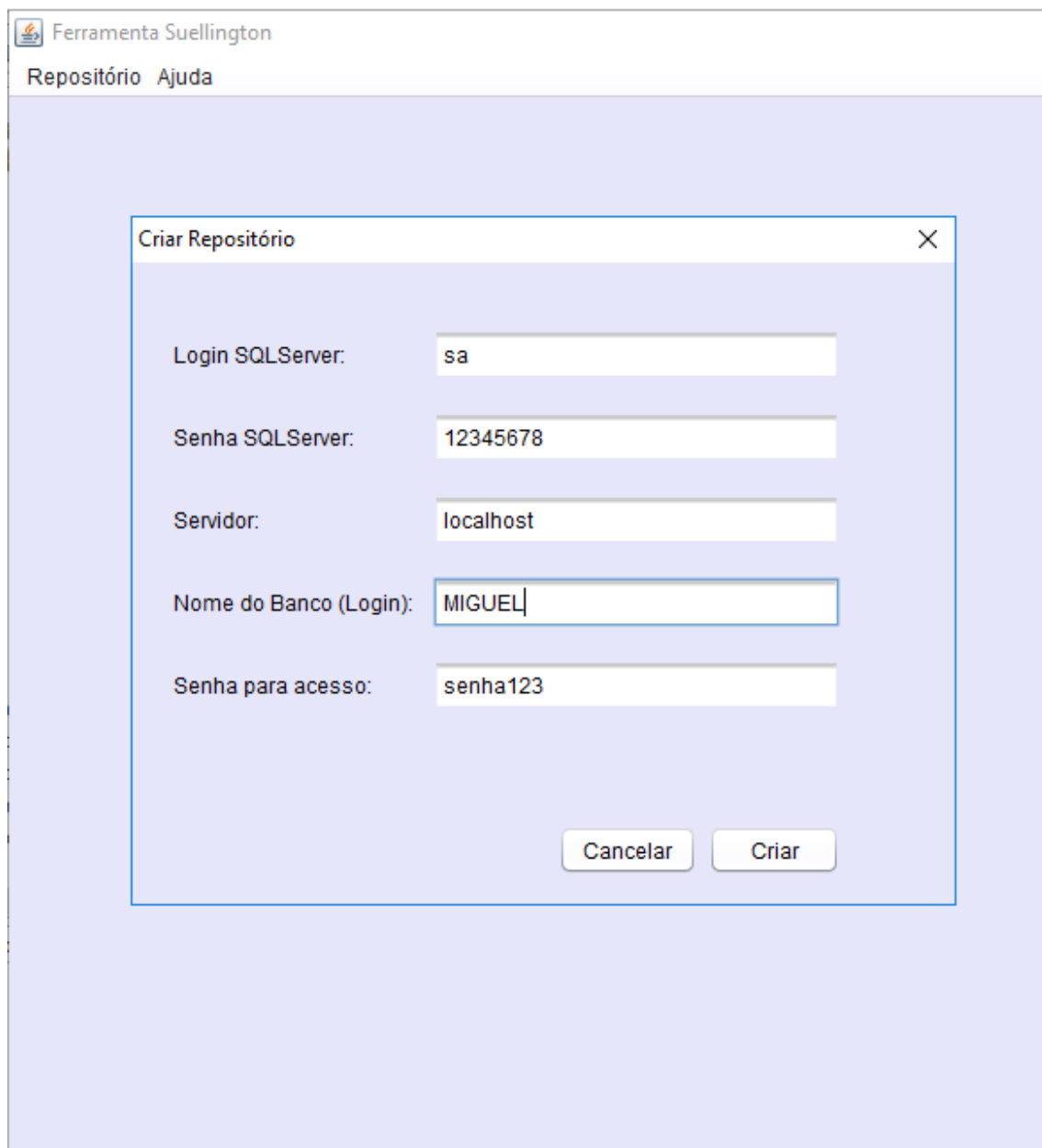
A Figura 23 apresenta a tela inicial da ferramenta. As opções do menu Repositório e Ajuda abre opções para iniciar o uso da ferramenta e para uma tela de informações básicas sobre a ferramenta, respectivamente.



**Figura 23 Tela Inicial da ferramenta**

### **Criar repositório**

A Figura 24 apresenta a tela para criar um repositório, opção que está no menu Repositório na barra de ferramentas. É necessário fornecer informações como Login SQLServer, Senha SQLServer, Servidor, Nome do banco, também usado como login e Senha para ter acesso à ferramenta.

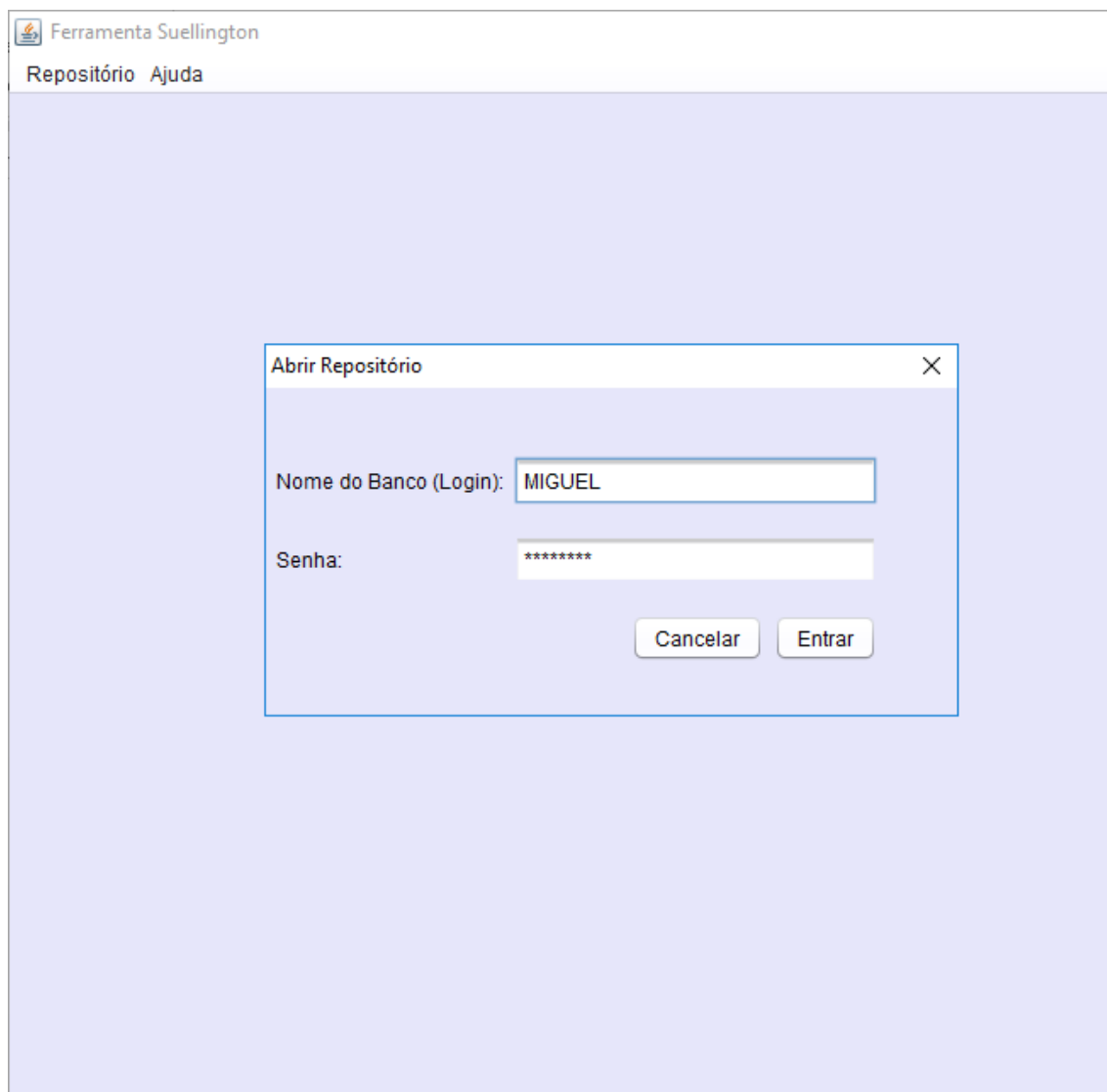


**Figura 24 Tela de Criar repositório**

### **Abrir Repositório**

A figura 25 apresenta a tela em que é usado um login, que é o mesmo nome do repositório e a senha criada para acesso e assim acessar o repositório para iniciar a configuração. A opção que está no menu Repositório na barra de ferramentas.

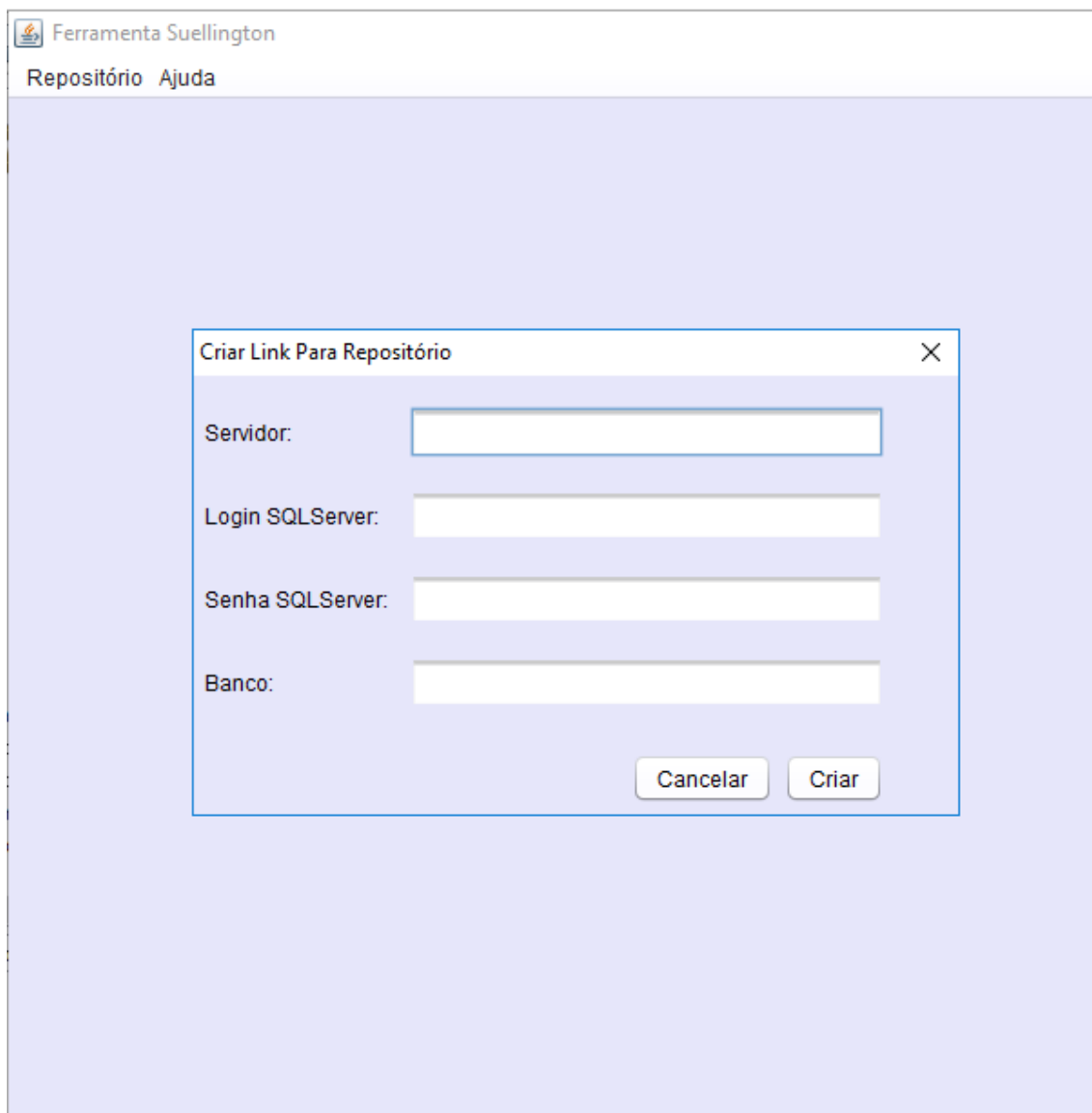




**Figura 25 Tela de Abrir Repositório**

### **Criar Link para Repositório**

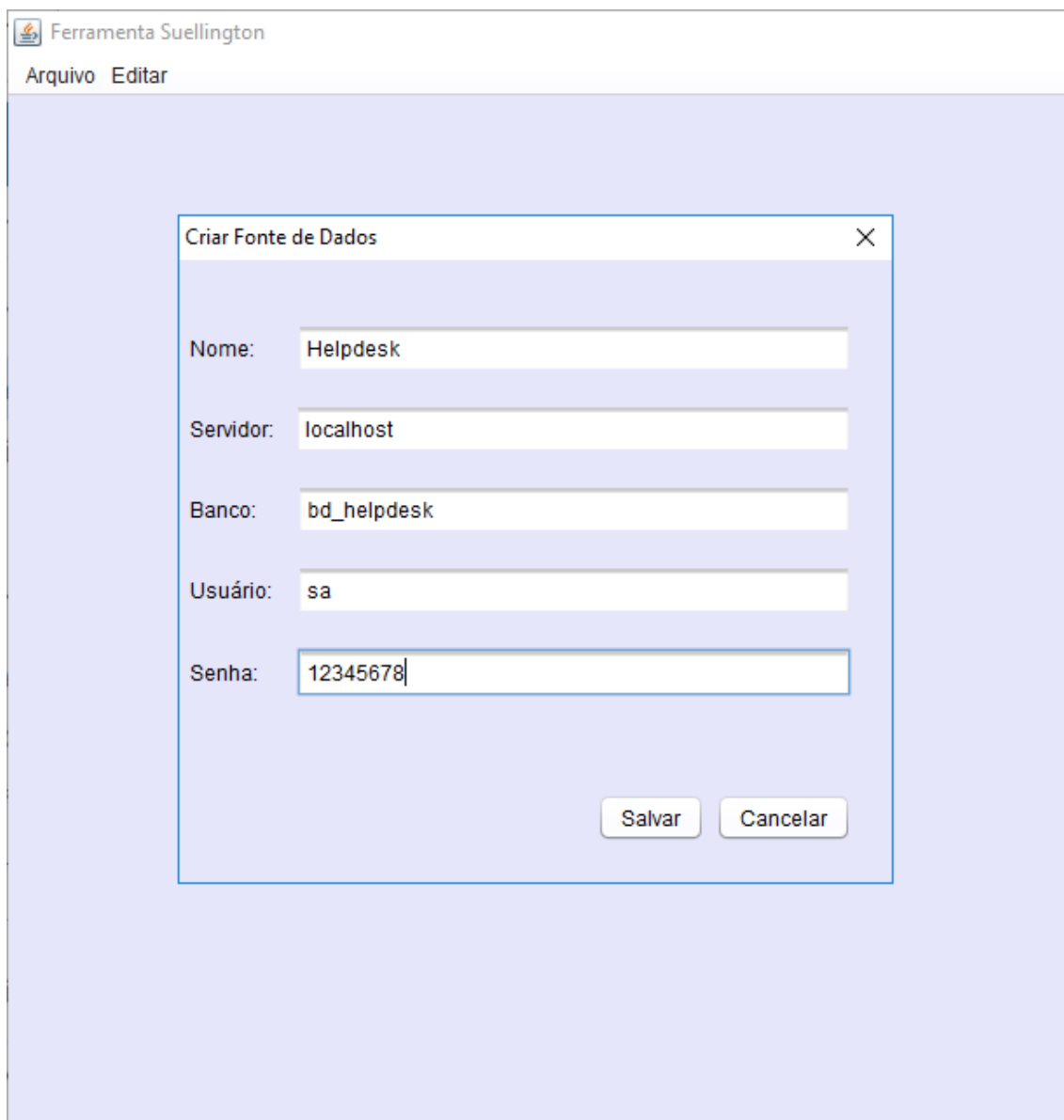
A Figura 26 apresenta a tela para criar um link para repositório, opção que está no menu Repositório na barra de ferramentas. É necessário fornecer informações como Servidor, Login SQLServer e o Nome do banco, também usado como login.



**Figura 26 Tela de Criar Link para Repositório**

### **Criar Fonte de Dados**

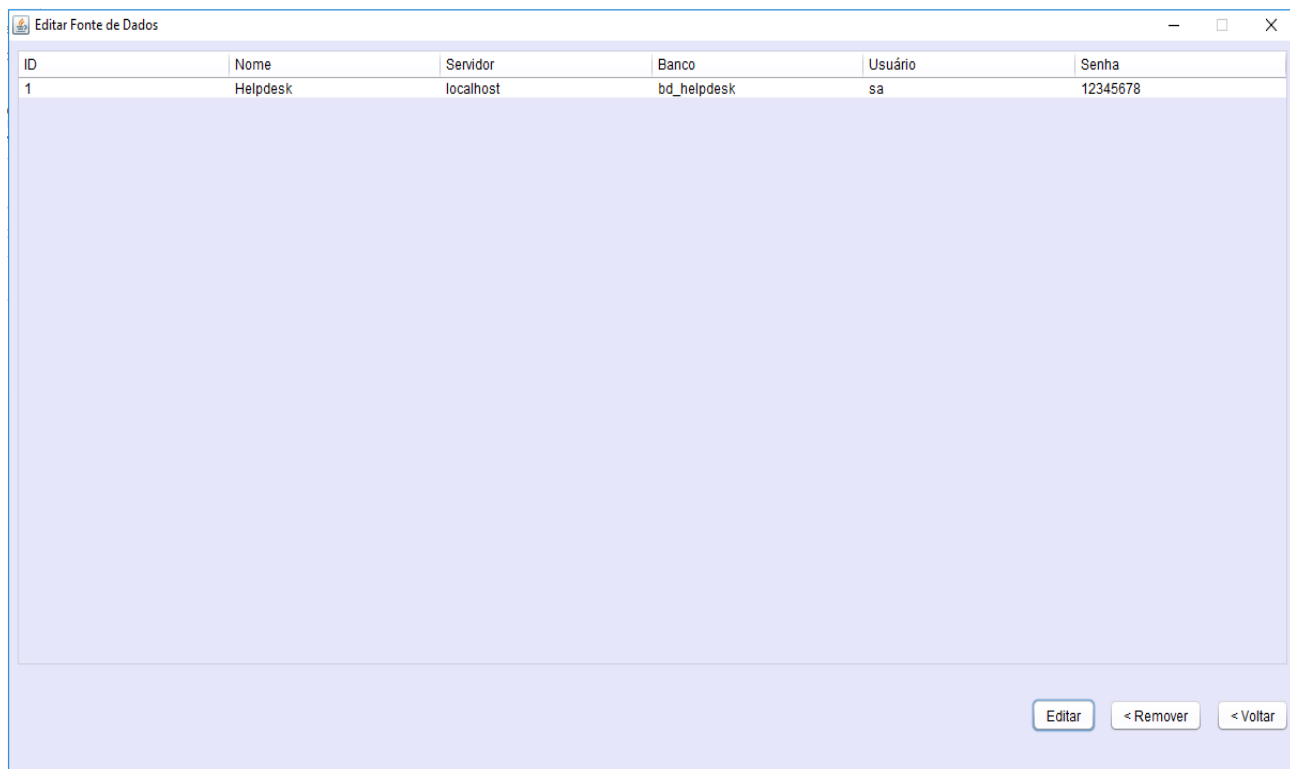
A Figura 27 apresenta a tela para criar uma fonte de dados, opção que está no menu Arquivo na barra de ferramentas depois que é efetuado o login. É necessário fornecer informações como Nome, Servidor, Banco, Usuário e Senha.



**Figura 27 Tela de Criar Fonte de Dados**

### **Editar Fonte de Dados**

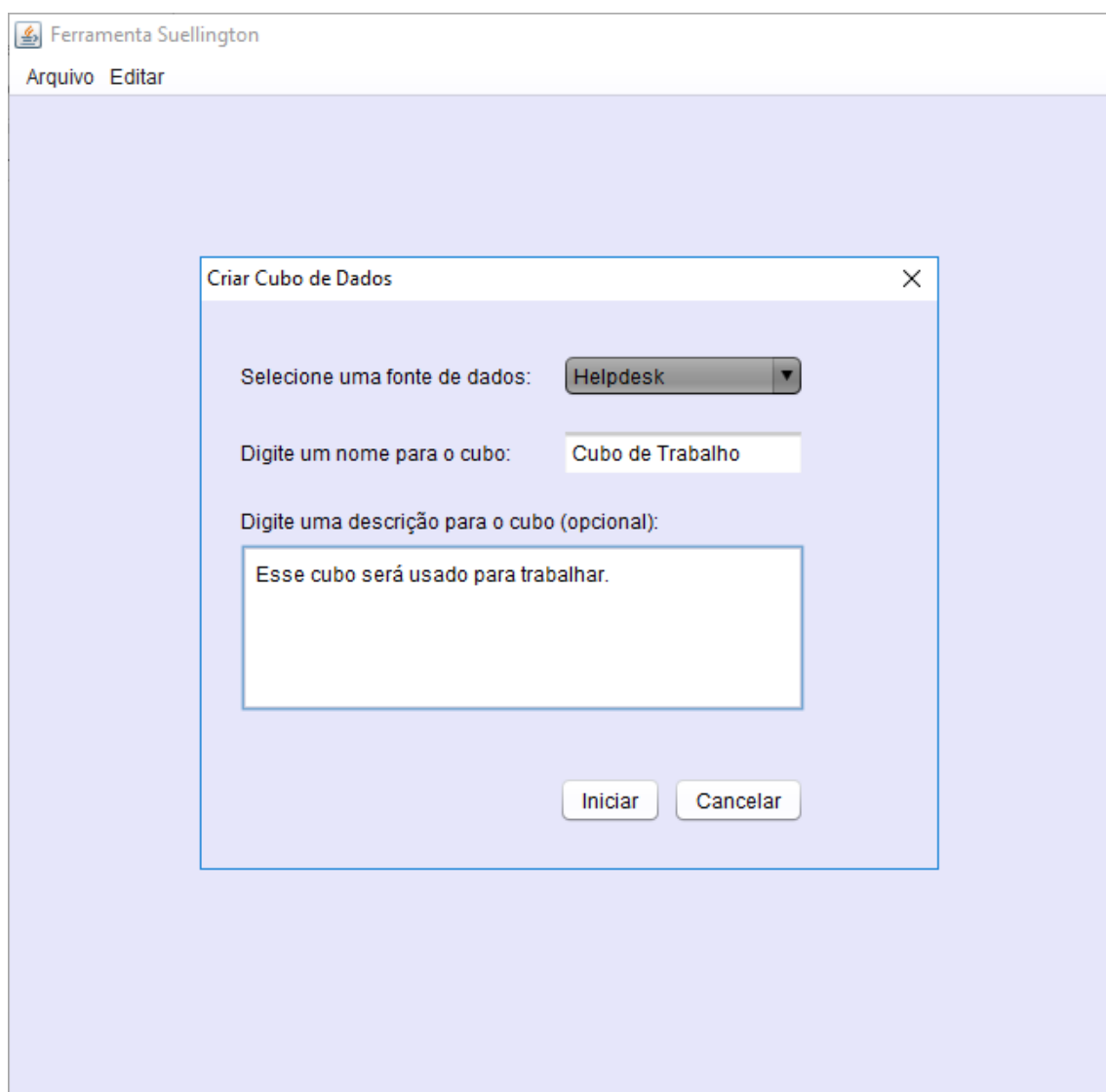
A Figura 28 apresenta a tela para editar ou remover uma fonte de dados, opção que está no menu Arquivo na barra de ferramentas depois que é efetuado o login. É necessário clicar sobre a fonte de dados que se deseja fazer uma das duas ações e então prosseguir.



**Figura 28 Tela de Editar ou Remover Fonte de Dados**

### **Criar Cubo de Dados**

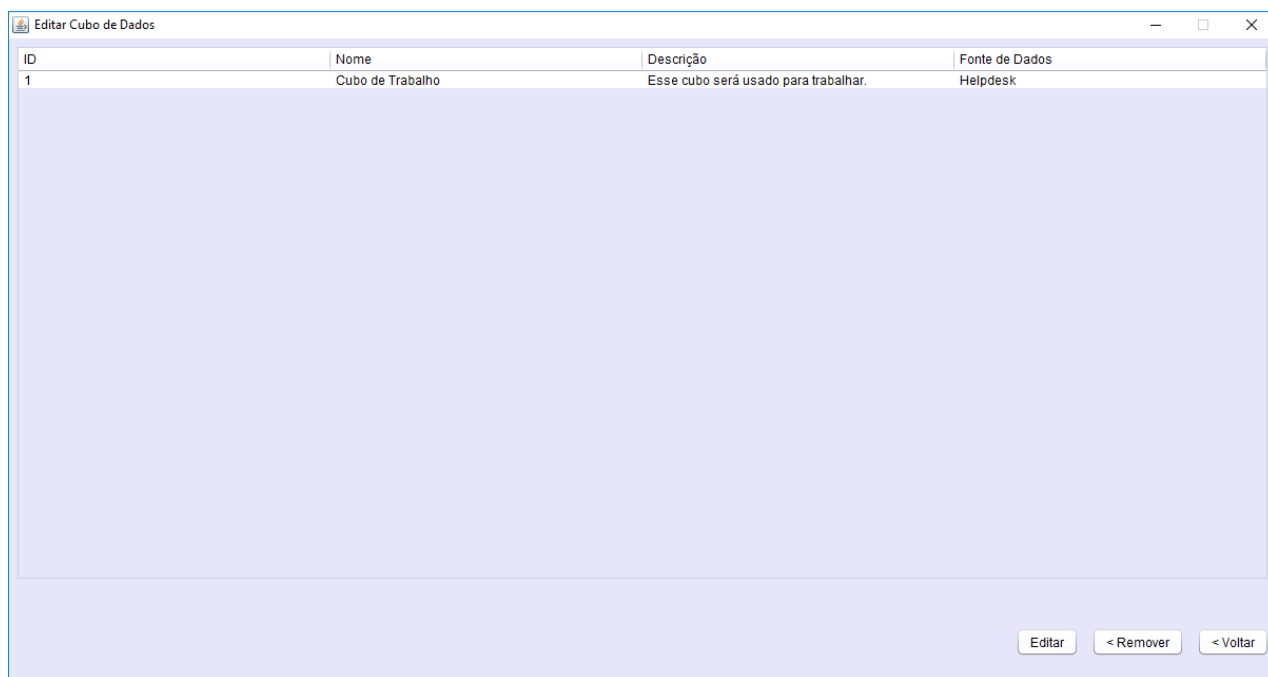
A Figura 29 apresenta a tela para criar um cubo de dados, opção que está no menu Arquivo na barra de ferramentas depois que é efetuado o login. É necessário fornecer informações como Nome, uma Descrição não obrigatória e selecionar uma Fonte de Dados que já deve ter sido cadastrada.



**Figura 29 Tela de Criar Cubo de Dados**

### **Editar Cubo de Dados**

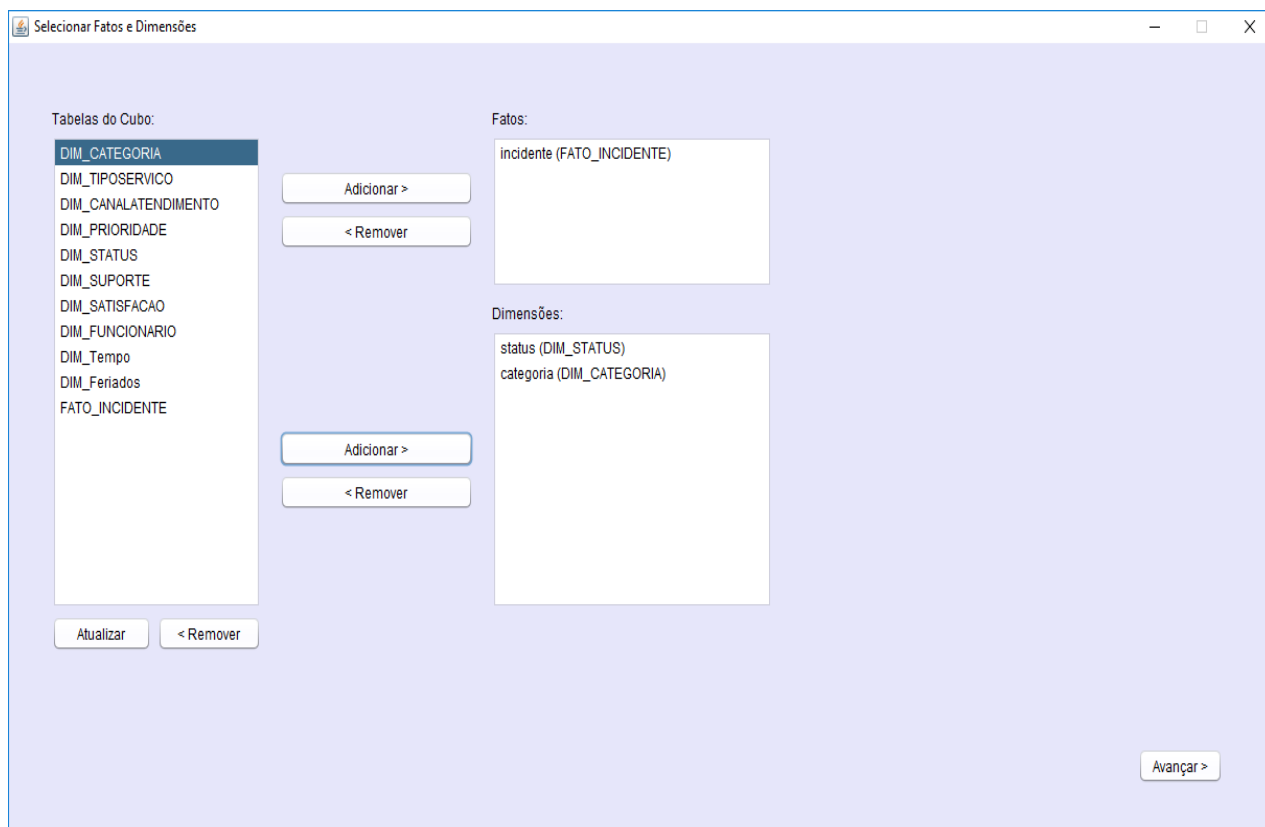
A Figura 30 apresenta a tela para editar ou remover um cubo de dados, opção que está no menu Arquivo na barra de ferramentas depois que é efetuado o login. É necessário clicar sobre o cubo de dados que se deseja fazer uma das duas ações e então prosseguir.



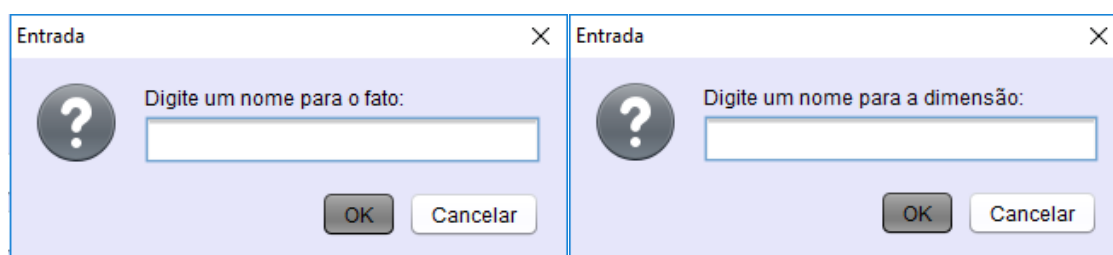
**Figura 30 Tela de Editar ou Remover Cubo de Dados**

### **Adicionar e remover tabela, adicionar e remover fato e adicionar e remover dimensão**

A Figura 31 apresenta a tela que permite adicionar e remover tabelas, funções que só são habilitadas quando se entra para editar o cubo, pois é com ela que pode remover uma tabela que não vai usar mais e pode adicionar uma tabela nova que foi criada. Também podemos ver as funcionalidades de adicionar e remover, tanto para o fato, quando para a dimensão, onde no momento de selecionar qual tabela é um fato ou uma dimensão é necessário informar uma descrição por umas das caixas de texto mostradas na figura 31. No lado esquerdo vemos as tabelas da fonte de dados e ao lado direito, em cima ficam as tabelas que foram identificadas como fato e embaixo as que foram identificadas como dimensão.



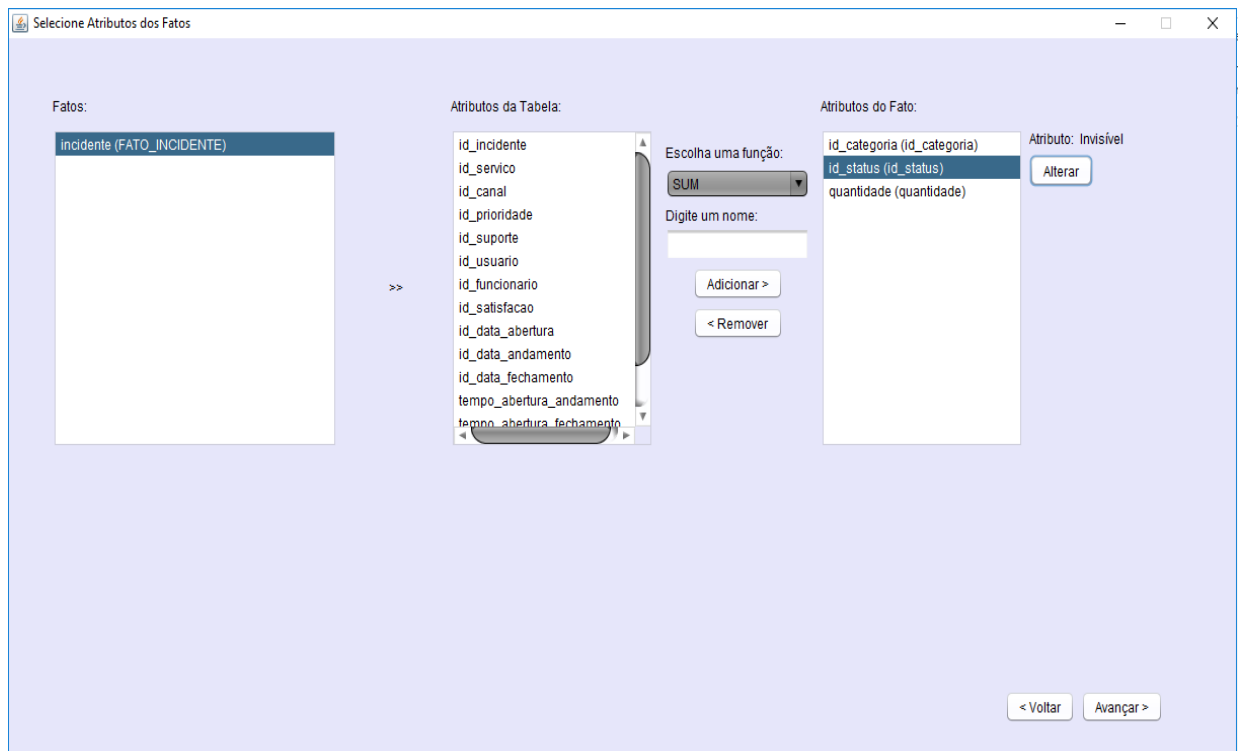
**Figura 31 Tela de adicionar e remover tabela, adicionar e remover fato e adicionar e remover dimensão**



**Figura 32 Caixa para adicionar um nome para o fato ou para a dimensão**

### **Adicionar e remover atributos do fato**

A Figura 33 apresenta a tela que permite adicionar e remover atributos do fato. Na primeira lista vemos o fato que foi escolhido no passo anterior, clicando nele vemos na segunda lista todos os atributos dele que ainda não foram escolhidos e na terceira caixa vemos o atributos que foram selecionados. Ainda podemos escolhe a função de agregação que irá acompanhar o atributo e podemos torna-lo invisível ao usuário através do botão Alterar depois que escolhe o atributo que quer fazer isso. Além de também ter o poder de dar o nome que desejar ao atributo.

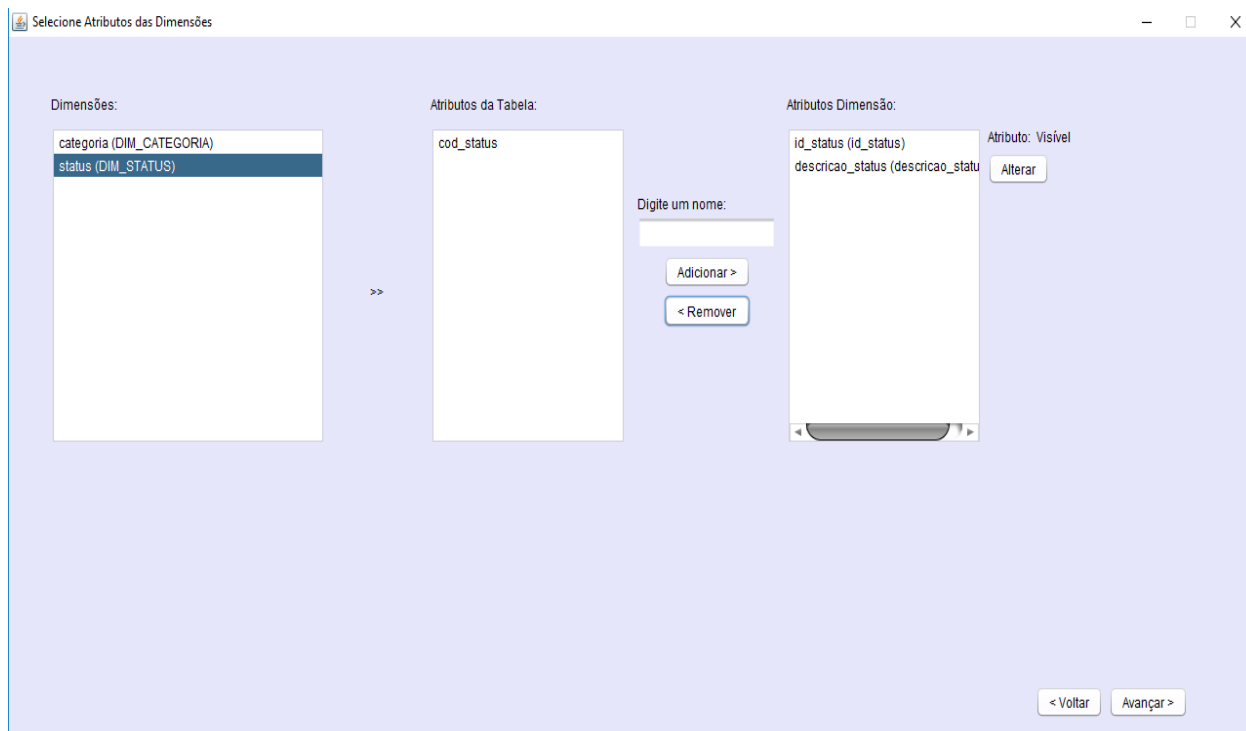


**Figura 33 Caixa para adicionar o atributo do fato**

### **Adicionar e remover atributos da dimensão**

A Figura 34 apresenta a tela que permite adicionar e remover atributos da dimensão. Na primeira lista vemos a dimensão que foi escolhida, clicando nela vemos na segunda lista todos os atributos dela que ainda não foram escolhidos e na terceira caixa vemos os atributos que foram selecionados. Ainda podemos torna-lo invisível ao usuário através do botão Alterar depois que escolhe o atributo que quer fazer isso. Além de também ter o poder de dar o nome que desejar ao atributo.

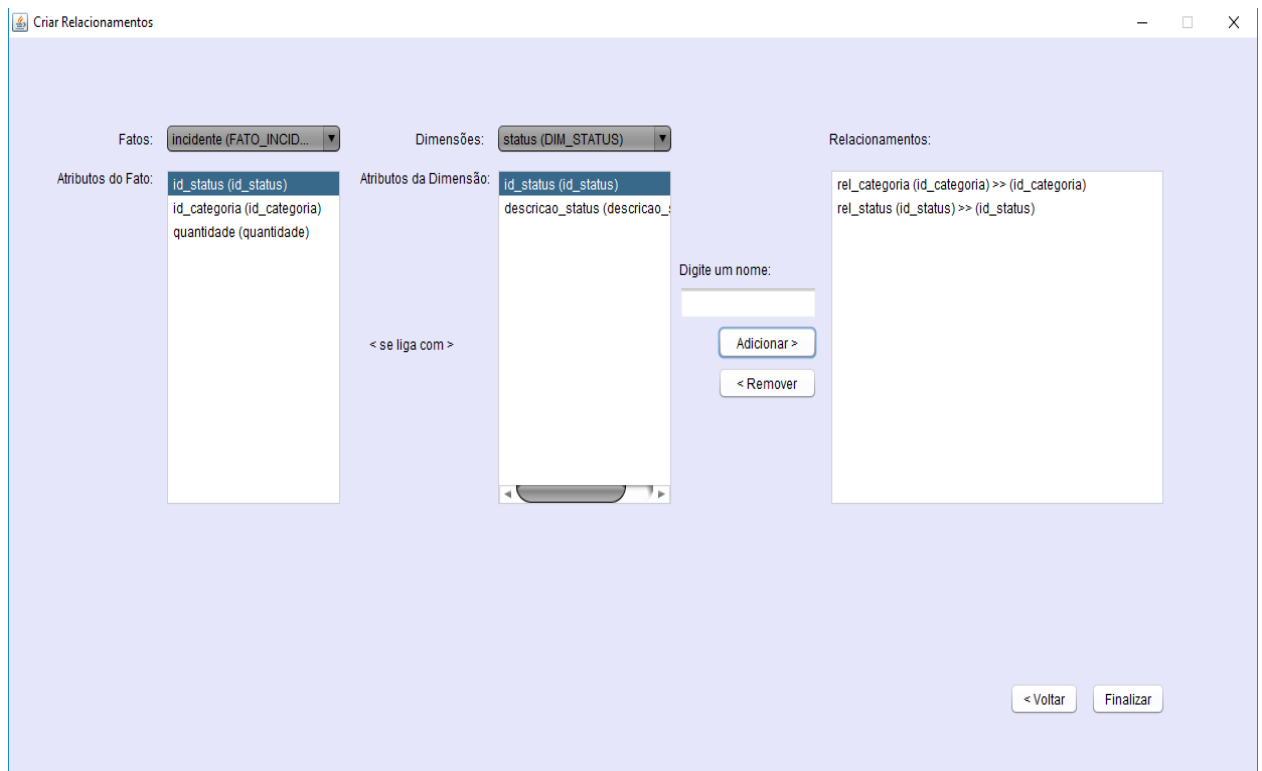




**Figura 34 Caixa para adicionar o atributo da dimensão**

### **Adicionar e remover relacionamento**

A Figura 35 apresenta a tela que permite adicionar e remover relacionamentos entre o fato e a dimensão, no combo acima de cada lista escolhe a tabela e logo abaixo aparecem seus atributos, sabendo como se ligam, clica em cada e adiciona, aparecendo assim na terceira lista. Lembrando que um nome precisa ser dado ao relacionamento através do campo de texto.



**Figura 35 Tela para adicionar e remover relacionamento**

## **5.2.Ferramenta de Usuário (Módulo de criação de consultas)**

### **Tela Inicial**

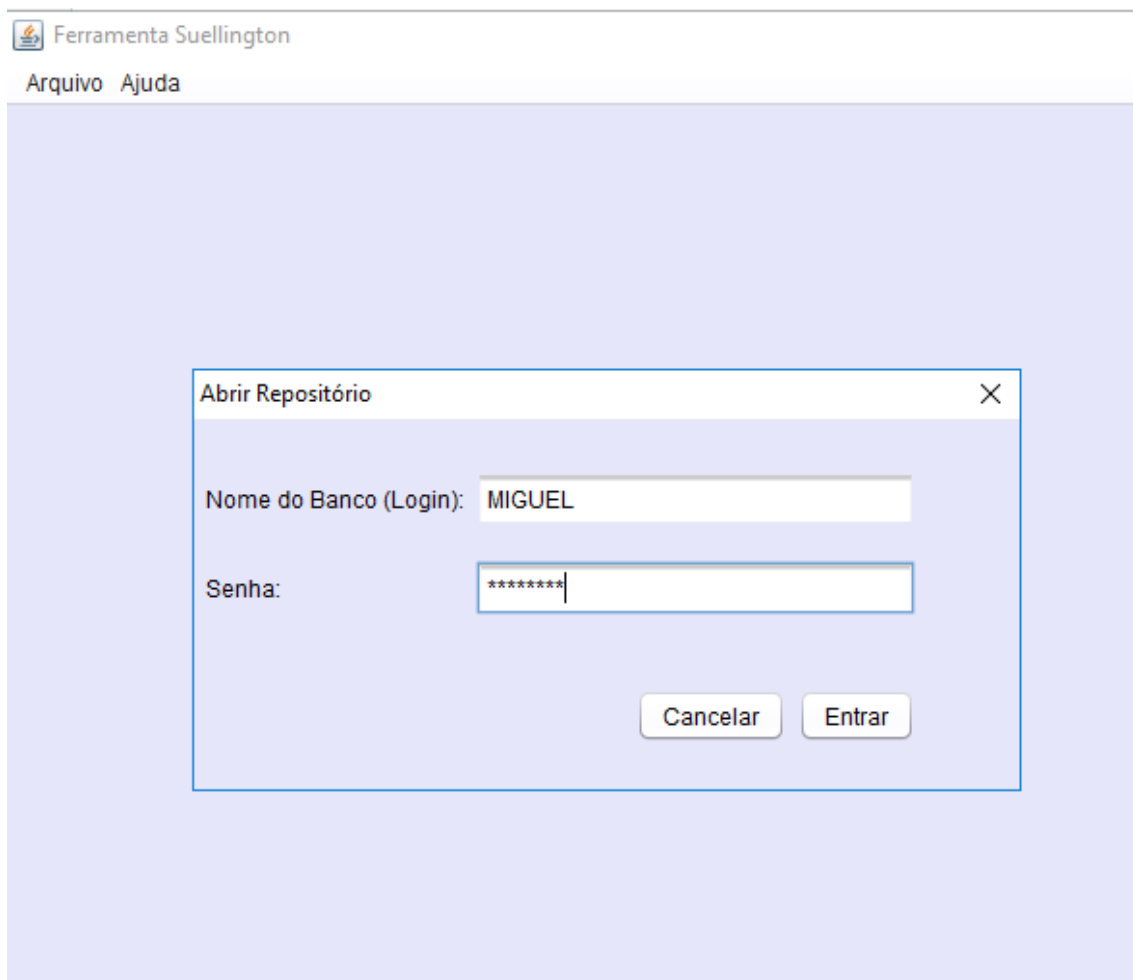
A Figura 36 apresenta a tela inicial da ferramenta. As opções do menu Arquivo e Ajuda abre opções para iniciar o uso da ferramenta e para uma tela de informações básicas sobre a ferramenta, respectivamente.



**Figura 36 Tela inicial**

## Abrir Repositório

A figura 37 apresenta a tela em que é usado um login, que é o mesmo nome do repositório e a senha criada para acesso e assim acessar o repositório para iniciar a configuração. A opção que está no menu Arquivo na barra de ferramentas.



**Figura 37 Abrir repositório**

## Selecionar cubo de dados

A figura 38 apresenta a tela em que se é escolhido o cubo de dados que será trabalhado.



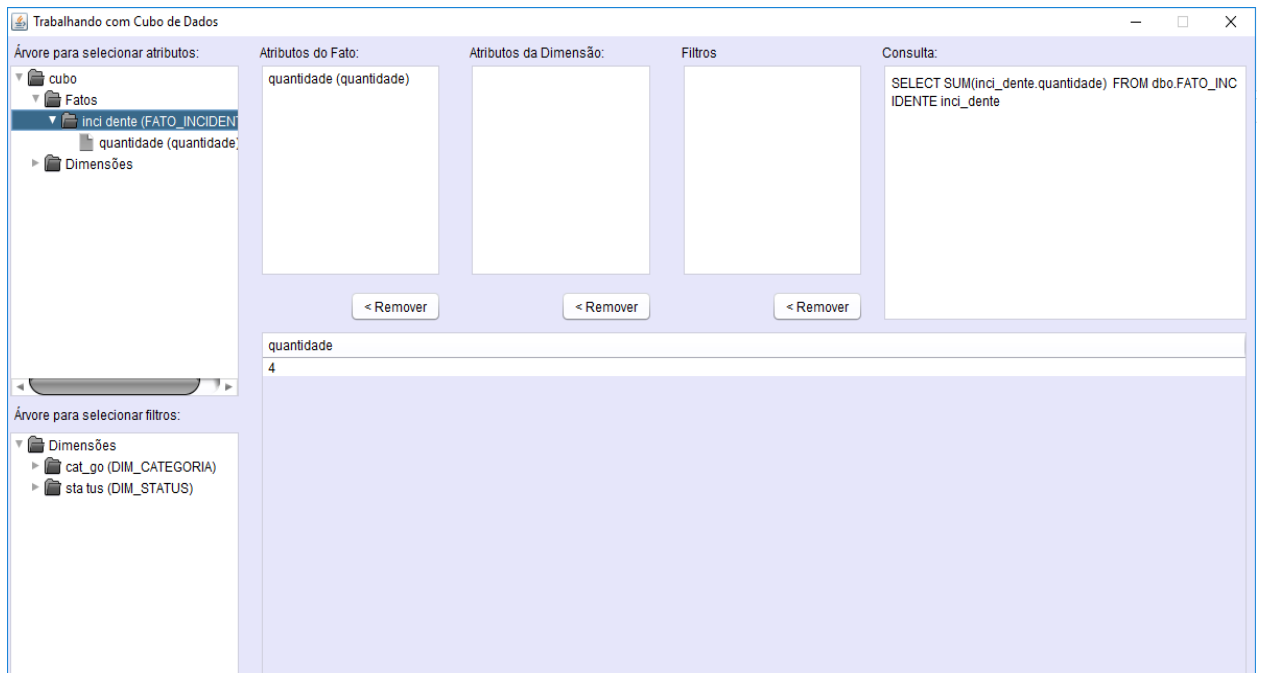
**Figura 38 Selecionar cubo de dados**

### **Criar consulta**

As figuras 39, 40 e 41 apresentam alguns exemplos na tela em que se é criada e executada a consulta, na árvore acima e à direita vemos o fato, as dimensões e seus atributos, onde clicando já o seleciona para a consulta que aparece no último campo de texto. Na segunda e terceira lista aparecem os atributos escolhidos para que possa fazer a remoção. Além disso, no espaço abaixo é mostrado o resultado da consulta que foi criada e executada automaticamente. Na árvore inferior à esquerda, podemos selecionar os atributos que irão se tornar cláusulas, tendo a opção de escolher o operador e digitar algo para a comparação, como mostrado na tela 42.



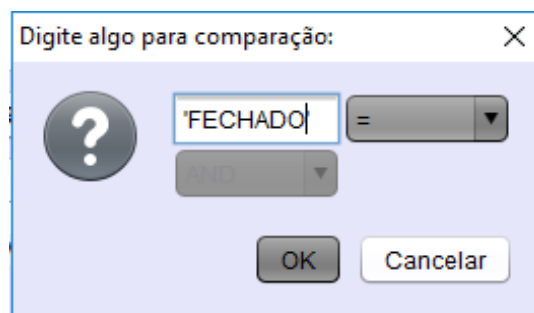
**Figura 39 Criar consulta 1**



**Figura 40 Criar consulta 2**



**Figura 41 Criar consulta 3**



**Figura 42 Caixa de texto para cláusulas**

## 6. CONCLUSÃO E TRABALHOS FUTUROS

O conhecimento sobre a arquitetura de ferramentas OLAP é crucial para os profissionais que trabalham na área de inteligência voltada para os negócios. Esse conhecimento facilita o processo de implementação, manutenção, migração para outras tecnologias e aquisição de produtos. Em particular, as ferramentas OLAP voltadas para a tecnologia relacional, também conhecidas como ROLAP, respondem por grande parte do mercado de ferramentas voltadas para ambientes de suporte à decisão. A maior dificuldade no estudo dessas tecnologias é que, embora tenham o mesmo propósito, suas particularidades mascaram os módulos que possuem em comum, dificultando um entendimento sobre a arquitetura da solução.

Com o objetivo de facilitar o ensino da arquitetura ROLAP, sem a necessidade de apresentar características particulares de uma ferramenta específica, esse trabalho criou um protótipo de solução ROLAP através da identificação de requisitos mínimos necessários para uma ferramenta dessa classe.

Composto por dois módulos principais, Administrador e Usuário, o protótipo apresenta, atualmente, características fundamentais de uma solução ROLAP: a) criação e gerenciamento de repositório; b) criação de camada multidimensional e mapeamento para esquemas relacionais; c) definição de funções de agregação; d) construção de consultas *ad-hoc* através da seleção de fatos, dimensões e filtros.

O protótipo foi avaliado de maneira contínua, pelo professor da disciplina de Sistemas de Suporte à Decisão, durante todo o desenvolvimento. As primeiras avaliações realizadas com o auxílio de exercícios de modelagem dimensional utilizados em sala de aula, além de terem sido cruciais para identificar requisitos antes não descobertos, apresentaram fortes indícios que a solução construída pode ser um forte aliado no ensino de ferramentas OLAP. No entanto, essas primeiras impressões precisam ser validadas através de experimentos que façam uso da ferramenta como instrumento. Faz-se necessário também simular um maior conjunto de casos de modelagem a fim de identificar os casos que a ferramenta ainda não suporta.

Como trabalhos futuros, temos duas linhas que podem ser seguidas: a primeira é a de experimentos para avaliação e validação da ferramenta como instrumento de ensino, e a segunda é a de evolução da solução para incluir características ainda não contempladas como, por exemplo, hierarquias explícitas, navegação em agregados, atributos de fato

calculados. Nessa linha de evolução, também é possível o desenvolvimento de interfaces clientes para *web* ou *mobile*.



## REFERÊNCIAS

ADAMSON, Christopher. **The complete reference star schema**. New York: McGraw-Hill, 2010.

COLAÇO JR., Methanias. **Projetando Sistemas de Apoio à Decisão Baseados em Datawarehouse**. Rio de Janeiro: Axcel Books, 2004.

CONNOLLY, T. M; BERG, C. E. **Database Systems: A Practical Approach to Design, Implementation, and Management (Global Edition)**, Pearson Education Limited, 2015.

COLONESE. G; MANHÃES. R. S; GONZÁLEZ. S. M; GALANTE. A. C; CARVALO. R. A; TANAKA. A. K; **Uma Ferramenta em Software Livre para o Desenvolvimento de Sistemas de Suporte à Decisão**. UNIRIO, Rio de Janeiro, 2008.

FERNANDES, M. S; KANTORSKI, G. Z. **Neuro BI: Uma Ferramenta Web Open Source Para O Apoio À Tomada De Decisão**, Santa Maria-RS, 2008.

KIMBALL, ROSS. **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)**, Wiley, 2002. ISBN 0471200247.

KIMBALL, ROSS. **A Dimensional Modeling Manifesto**. Disponível em: <<http://www.kimballgroup.com/1997/08/a-dimensional-modeling-manifesto/>> Acesso em 24 de Agosto de 2017

INMON, W.H. **Building the Data Warehouse, 4th edition**. New York City: John Wiley & Sons, Inc., 2005. 543 p.

Little, J. D. C. **Models and Managers: The Concept of a Decision Calculus**. Management Science, vol. 16, no. 8, pp. B466-485, April 1970.